Title

Change Point Detection for Dynamic Graphs and Dynamic Valued Networks Modeling

Author

Kei, Yik Lun

Publication Date

2024

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

Change Point Detection for Dynamic Graphs and Dynamic Valued Networks Modeling

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Statistics

by

Yik Lun Kei

2024

ABSTRACT OF THE DISSERTATION

Change Point Detection for Dynamic Graphs and
Dynamic Valued Networks Modeling

by

Yik Lun Kei
Doctor of Philosophy in Statistics
University of California, Los Angeles, 2024
Professor Oscar Hernan Madrid Padilla, Chair

Networks or graphs are often used to represent relational phenomena in numerous domains, and relational phenomena by nature progress in time. Devising powerful models and tools for dynamic graphs can provide valuable insights into real-world phenomena that benefit decision-making. Moreover, change point detection plays an indispensable role in identifying discrepancies in the data generating processes. Without taking the structural changes across dynamic networks into account, learning from the time series may lead to ambiguity. Thence, it is practical for researchers to first localize the change points, and then analyze the dynamic networks, rather than neglecting where the network patterns have substantially changed.

We first consider the change point detection problem for dynamic graphs using the Separable Temporal Exponential-family Random Graph Model (STERGM). The STERGM that utilizes network statistics to represent the network structures is a flexible model to fit dynamic graphs. We propose a new estimator derived from the Alternating Direction Method of Multipliers (ADMM) and Group Fused Lasso to simultaneously detect multiple time points,

where the parameters of a time-heterogeneous STERGM have changed.

Then we study the change point detection problem for dynamic graphs under a generative framework. The proposed model consists of learnable prior distributions for graph-level representations and of a decoder that can generate dynamic graphs from the low-dimensional representations. The informative prior distributions in the latent spaces are learned from the observed graphs as empirical Bayes, and the expressive power of a generative model is exploited to assist change point detection.

Furthermore, we consider an exponential-family model to fit dynamic valued networks, as relations by nature often have degree of strength. To facilitate the modeling of dyad value increment and decrement, a Partially Separable Temporal Exponential-family Random Graph Model is proposed. The parameter learning algorithms approximate the maximum likelihood, by drawing Markov chain Monte Carlo (MCMC) samples conditioning on the valued network from the previous time step.

Throughout the dissertation, we use both simulated and real-world data to evaluate the methodology and the learning algorithms. The results demonstrate the effectiveness of the proposed frameworks.

The dissertation of Yik Lun Kei is approved.

Ying Nian Wu

Jingyi Li

Mark Stephen Handcock

Oscar Hernan Madrid Padilla, Committee Chair

University of California, Los Angeles

2024

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to Professor Oscar Hernan Madrid Padilla for advising me throughout my doctoral studies. His broad knowledge of statistical theory always leads to an innovative solution for a challenging problem, and his deep knowledge of optimization always guides me to produce meaningful results through practical methods. I am indebted to his belief in my potential, his encouragement during my moments of doubt, and his insights into my research. I am immensely fortunate to have such a remarkable advisor by my side, and I will carry his guidance with me as I embark on my next academic and professional endeavors.

I would also like to thank Professor Mark Handcock for the extensive discussion on network modeling and for having me in his reading group during my doctoral studies. The STERGM studied in this dissertation is an innovation of Mark Handcock, and without it, many works in this dissertation would not have been possible. Moreover, I would also like to thank other committee members for their guidance and support, and I would like to thank my friends for their company and collaboration.

Lastly, I would like to thank MH for always being by my side. MH's unwavering support and encouragement have been instrumental in shaping my character. MH's impact on my life is immeasurable and I am eternally grateful for everything MH has done.

VITA

2019-2024	Teaching Fellow, Department of Statistics, UCLA
2023	Data Scientist Intern, Amazon
2017-2019	Senior Data Analyst, UCLA Health
2012-2017	B.S. in Statistics & B.A. in Economics, UCLA

PUBLICATIONS

Change Point Detection in Dynamic Graphs with Generative Model, Yik Lun Kei, Jialiang Li, Hangjian Li, Yanzhen Chen, Oscar Hernan Madrid Padilla, Under Review

Change Point Detection on a Separable Model for Dynamic Networks, Yik Lun Kei, Hangjian Li, Yanzhen Chen, Oscar Hernan Madrid Padilla, Under Review

A Partially Separable Model for Dynamic Valued Networks, Yik Lun Kei, Yanzhen Chen, Oscar Hernan Madrid Padilla, Computational Statistics & Data Analysis, 2023

CHAPTER 1

Introduction

Networks are often used to describe relational phenomena that cannot be reduced merely to the attributes of individuals. The abundance and the complexity of network data in the real world demand statistical models and methodologies to dissect and understand the relational phenomena. Moreover, relational phenomena by nature can progress over time, and a pivotal aspect of understanding the dynamics is the identification of change points where the underlying network structure fundamentally changes. In brief, this dissertation is structured around three self-contained articles [KCP23, KLC23, KLL24], contributing to the fields of change point detection and dynamic network modeling.

The second chapter introduces an approach to detect change points in dynamic networks using the Separable Temporal Exponential-family Random Graph Model (STERGM). This model manages dyad formation and dyad dissolution separately, to capture the structural changes in network evolution realistically. The flexibility of STERGM and the extensive selection of network statistics also boost the power of the proposed method. Essentially, we fit a time-heterogeneous STERGM to the dynamic networks, while penalizing the sum of Euclidean norms of the parameter differences between consecutive time steps. The objective function that consists of the negative log-likelihood and the Group Fused Lasso regularization is minimized via the Alternating Direction Method of Multipliers (ADMM), and we adopt the pseudo-likelihood of STERGM to expedite the estimation process.

The third chapter delves into the usage of generative models to detect change points in dynamic graphs. Inherently, dynamic networks can be complex due to both dyadic and

temporal dependencies. Learning low-dimensional graph representations can extract useful patterns from the observed data to facilitate the change point detection task. The proposed framework utilizes learnable prior distributions and a graph decoding mechanism to capture the structural changes. Specifically, the informative priors for the graph representations in the latent space are learned from the observed data as empirical Bayes, and the model parameters are learned via maximum approximate likelihood with a Group Fused Lasso regularization. The resulting optimization problem is solved via ADMM, and the generative model is demonstrated to be useful for change point detection.

The fourth chapter proposes a Partially Separable Temporal Exponential-family Random Graph Model (PST ERGM) to fit dynamic valued networks. Conventionally, a relation between two actors is indicated by the presence or absence of a tie: a binary state between two nodes in a network. Though connected ties are seemingly identical in binary networks, relations by nature often have degrees of strength, which can be represented by generic values to distinguish them. Transitioning from the dichotomous state of relation to the granular magnitude of strength requires a class of models to comprehend the richness of valued networks. In particular, the proposed model assumes the factors that increase relational strength are different from those that decrease relational strength. Therefore, we can construct two intermediate networks to manage dyad value increment and decrement separately. The dynamics are specified with two sets of network statistics evaluated on the intermediate networks, and we can use two sets of parameters to facilitate interpretation.

In summary, this dissertation contributes to the understanding of dynamic networks through statistical models and computational techniques. By utilizing both simulated and real-world data, the results demonstrate the effectiveness of these proposed frameworks.

CHAPTER 2

A Separable Model for Change Point Detection in Dynamic Graphs

This chapter studies change point detection in time series of networks, with the Separable Temporal Exponential-family Random Graph Model (STERGM). Dynamic network patterns can be inherently complex due to dyadic and temporal dependence. Detection of the change points can identify the discrepancies in the underlying data generating processes and facilitate downstream analysis. The STERGM that utilizes network statistics to represent the structural patterns is a flexible model to fit dynamic networks. We propose a new estimator derived from the Alternating Direction Method of Multipliers (ADMM) and Group Fused Lasso to simultaneously detect multiple time points, where the parameters of a time-heterogeneous STERGM have changed. We also provide a Bayesian information criterion for model selection and an R package `CPDstergm` to implement the proposed method. Experiments on simulated and real data show good performance of the proposed framework.

2.1 Introduction

Networks are often used to describe relational phenomena that cannot be limited merely to the attributes of individuals. In an investigation of the transmission of COVID-19, [FDR21] used networks to represent human mobility and forecast disease incidents. The study of physical connections, beyond the health status of individuals, permits policymakers to implement preventive measures effectively and allocate healthcare resources efficiently. Yet

relations can change over time, and dynamic relational phenomena are often aggregated into a static network for analysis. To this end, a temporal model for dynamic networks is in high demand.

In recent decades, a plethora of models has been proposed for dynamic networks analysis. [Sni01], [Sni05], and [SBS10] developed a Stochastic Actor-Oriented Model, which is driven by the actor's perspective to make or withdraw ties to other actors in a network. [KSA10] focused on recovering the latent time-varying graph structures of Markov Random Fields, from serial observations of nodal attributes. [SM05], [SC15], and [SC16] presented a Latent Space Model, by assuming the edges between actors are more likely when they are closer in the latent Euclidean space. [MM17], [LEN18], and [Pen19] investigated the dynamic Stochastic Block Model (SBM), and [JLY20] developed an Autoregressive SBM to characterize the communities. Furthermore, the Exponential-family Random Graph Model (ERGM) that uses local forces to shape global structures [HHB08a] is a promising model for networks with dependent ties. [HFX10] defined a Temporal ERGM (TERGM), by conditioning on previous networks in the network statistics of an ERGM. [DC12b] proposed a bootstrap approach to maximize the pseudo-likelihood of the TERGM and assess uncertainty. In general, network evolution concerns the rate at which edges form and dissolve. Demonstrated in [KH14], these two factors can be mutually interfering, making the dynamic models used in the literature difficult to interpret. Posing that the underlying reasons that result in dyad formation are different from those that result in dyad dissolution, [KH14] proposed a Separable TERGM (STERGM) to dissect the entanglement with two conditionally independent models.

In time series analysis, change point detection plays a central role in identifying the discrepancies in the underlying data generating processes. Without taking the structural changes across dynamic networks into consideration, learning from the time series may not be meaningful. As relational phenomena are studied in numerous domains, it is practical for researcher to first localize the change points, and then analyze the dynamic networks, rather than overlooking where the network patterns have substantially changed.

There has also been an increasing interest in studying the change point detection problem for dynamic networks. [WTP13] focused on the Stochastic Block Model time series, and [WYR21] studied a sequence of inhomogeneous Bernoulli networks. [LBF21], [MBF22], and [PYP22] considered a sequence of Random Dot Product Graphs that are both dyadic and temporal dependent. Methodologically, [CZ15] and [CC19] developed a graph-based approach to delineate the distributional differences before and after a change point, and [Che19] utilized the nearest neighbor information to detect the changes in an online framework. [ZCL19] proposed a two-step approach that consists of an initial graphon estimation followed by a screening algorithm, [SC22c] exploited the features in high dimensions via a kernel-based method, and [CAA20] employed embedding methods to detect both anomalous graphs and anomalous vertices. Moreover, [LCX18] introduced an eigenvector-based method to reveal the change and persistence in the gene communities for a developing brain. [BA18] focused on a Gaussian Graphical Model to detect the change points in the covariance structure of the Standard and Poor's 500. [OOC21] proposed a factorized binary search method to understand brain connectivity from the functional Magnetic Resonance Imaging time series data.

Allowing for user-specified network statistics to determine the structural changes for the detection, we make the following contributions in the proposed framework:

- To simultaneously detect multiple change points from a sequence of networks, we learn a time-heterogeneous STERGM, while penalizing the sum of Euclidean norms of the sequential parameter differences. We formulate the augmented Lagrangian as a Group Fused Lasso problem, and we derive an Alternating Direction Method of Multipliers (ADMM) to solve the optimization problem.

- We exploit the practicality of STERGM, which manages dyad formation and dissolution separately, to capture the structural changes in network evolution. The flexibility of STERGM, which considers both dyadic and temporal dependence, and the extensive

selection of network statistics also boost the power of the proposed method. Moreover, we provide a Bayesian information criterion for model selection, and we develop an R package `CPDstergm` to implement the proposed method.

- We simulate dynamic networks to imitate realistic social interactions, and our method can achieve greater accuracy on the networks that are both dyadic and temporal dependent. Furthermore, we punctually detect the winter and spring vacations with the MIT cellphone data [EP06]. We also detect three major change points from the stock market data analyzed in [JM15], and the detected change points align with three significant events during the 2008 worldwide economic crisis.

The rest of the chapter is organized as follows. In Section 2.2, we review the STERGM for dynamic networks. In Section 2.3, we present the likelihood-based objective function with Group Fused Lasso regularization, and we derive an ADMM to solve the optimization problem. In Section 2.4, we discuss change points localization after parameter learning, along with model selection and post-processing. In Section 2.5, we implement our method on simulated and real data. In Section 2.6, we conclude our work with a discussion and potential future developments.

2.2 STERGM Change Point Model

2.2.1 Notation

For a matrix $\boldsymbol{X} \in \mathbb{R}^{\tau \times p}$, denote $\boldsymbol{X}_{i,\cdot} \in \mathbb{R}^{1 \times p}$ and $\boldsymbol{X}_{\cdot,i} \in \mathbb{R}^{\tau \times 1}$ as the respective ith row and ith column of the matrix $\boldsymbol{X}$. Moreover, denote $\boldsymbol{X}_{-i,\cdot} \in \mathbb{R}^{\tau \times p}$ as the matrix obtained by replacing the ith row of the matrix $\boldsymbol{X}$ with a zero vector, and $\boldsymbol{X}_{\cdot,-i} \in \mathbb{R}^{\tau \times p}$ is denoted similarly.

For a matrix $\boldsymbol{\theta} \in \mathbb{R}^{\tau \times p}$, define the transformation from a matrix to a vector as $\vec{\boldsymbol{\theta}} = \mathrm{vec}_{\tau p}(\boldsymbol{\theta}) \in \mathbb{R}^{\tau p \times 1}$, by sequentially concatenating each row of $\boldsymbol{\theta}$ to construct the vector $\vec{\boldsymbol{\theta}}$.

Reversely, for a vector $\vec{\boldsymbol{\theta}} \in \mathbb{R}^{\tau p \times 1}$, define the transformation from a vector to a matrix as $\boldsymbol{\theta} = \text{vec}_{\tau,p}^{-1}(\vec{\boldsymbol{\theta}}) \in \mathbb{R}^{\tau \times p}$, by sequentially folding the vector $\vec{\boldsymbol{\theta}}$ for every p elements into a row to construct the matrix $\boldsymbol{\theta}$.

2.2.2 ERGM

For a node set $N = \{1, 2, \cdots, n\}$, we can use a network $\boldsymbol{y} \in \mathcal{Y}$ to represent the potential relations for all pairs $(i, j) \in \mathbb{Y} \subseteq N \times N$. The network $\boldsymbol{y}$ has dyad $\boldsymbol{y}_{ij} \in \{0, 1\}$ to indicate the absence or presence of a relation between node i and node j, and $\mathcal{Y} \subseteq 2^{\mathbb{Y}}$. Moreover, we prohibit a network to have self-edge, so the diagonal elements of the network $\boldsymbol{y}$ are zeros. The relations in a network can be either directed or undirected, where an undirected network has $\boldsymbol{y}_{ij} = \boldsymbol{y}_{ji}$ for all (i, j).

The probabilistic formulation of an ERGM is

$$P(\boldsymbol{y}; \boldsymbol{\theta}) = \exp[\boldsymbol{\theta}^\top \boldsymbol{g}(\boldsymbol{y}) - \psi(\boldsymbol{\theta})] \tag{2.1}$$

where $\boldsymbol{g}(\boldsymbol{y})$, with $\boldsymbol{g} : \mathcal{Y} \to \mathbb{R}^p$, is a vector of network statistics; $\boldsymbol{\theta} \in \mathbb{R}^p$ is a vector of parameters; $\exp[\psi(\boldsymbol{\theta})] = \sum_{\boldsymbol{y} \in \mathcal{Y}} \exp[\boldsymbol{\theta}^\top \boldsymbol{g}(\boldsymbol{y})]$ is the normalizing constant. The network statistics $\boldsymbol{g}(\boldsymbol{y})$ may depend on nodal attributes $\boldsymbol{x}$. For notational simplicity, we omit the dependence of $\boldsymbol{g}(\boldsymbol{y})$ on $\boldsymbol{x}$.

With a surrogate as in [Bes74, SI90, VGH09, DC12b, HHH12a], the log-likelihood of an ERGM in (2.1) can be approximated as

$$l(\boldsymbol{\theta}) = \sum_{(i,j) \in \mathbb{Y}} \boldsymbol{y}_{ij} [\boldsymbol{\theta} \cdot \Delta \boldsymbol{g}_{ij}(\boldsymbol{y})] - \log \left\{ 1 + \exp[\boldsymbol{\theta} \cdot \Delta \boldsymbol{g}_{ij}(\boldsymbol{y})] \right\}$$

where the change statistics $\Delta \boldsymbol{g}_{ij}(\boldsymbol{y}) \in \mathbb{R}^p$ denote the change in $\boldsymbol{g}(\boldsymbol{y})$ when $\boldsymbol{y}_{ij}$ changes from 0 to 1, while rest of the network remains the same. This formulation is called the logarithm of the pseudo-likelihood, and it is helpful in ERGM parameter estimation. Next we introduce the Separable Temporal ERGM (STERGM) used in our change point model.

2.2.3 STERGM

For a sequence of networks, network evolution concerns (1) incidence: how often new ties are formed, and (2) duration: how long old ties last since they were formed. [DH18], [GD20], and [JLY20] pointed out that modeling snapshots of networks may give limited information about the transitions. To address this issue, [KH14] designed two intermediate networks, formation and dissolution networks, to reflect the incidence and duration. In particular, the incidence can be measured by dyad formation, and the duration can be traced by dyad dissolution.

Let $\boldsymbol{y}^t \in \mathcal{Y}^t \subseteq 2^{\mathbb{Y}}$ be a network observed at a discrete time point t. The formation network $\boldsymbol{y}^{+,t} \in \mathcal{Y}^{+,t}$ is obtained by attaching the edges that formed at time t to $\boldsymbol{y}^{t-1}$, and $\mathcal{Y}^{+,t} \subseteq \{\boldsymbol{y} \in 2^{\mathbb{Y}} : \boldsymbol{y} \supseteq \boldsymbol{y}^{t-1}\}$. The dissolution network $\boldsymbol{y}^{-,t} \in \mathcal{Y}^{-,t}$ is obtained by deleting the edges that dissolved at time t from $\boldsymbol{y}^{t-1}$, and $\mathcal{Y}^{-,t} \subseteq \{\boldsymbol{y} \in 2^{\mathbb{Y}} : \boldsymbol{y} \subseteq \boldsymbol{y}^{t-1}\}$. We also use the notation from [KCP23] to specify the respective formation and dissolution networks between time $t-1$ and time t for a dyad (i,j) as $\boldsymbol{y}_{ij}^{+,t} = \max(\boldsymbol{y}_{ij}^{t-1}, \boldsymbol{y}_{ij}^{t})$ and $\boldsymbol{y}_{ij}^{-,t} = \min(\boldsymbol{y}_{ij}^{t-1}, \boldsymbol{y}_{ij}^{t})$. In summary, $\boldsymbol{y}^{+,t}$ and $\boldsymbol{y}^{-,t}$ incorporate the dependence on $\boldsymbol{y}^{t-1}$ through construction, and they can be considered as two latent networks recovered from both $\boldsymbol{y}^{t-1}$ and $\boldsymbol{y}^{t}$ to emphasize the transition from time $t-1$ to time t.

Posing that the underlying factors that result in edge formation are different from those that result in edge dissolution, [KH14] proposed the STERGM to dissect the evolution between consecutive networks. Assuming $\boldsymbol{y}^{+,t}$ is conditionally independent of $\boldsymbol{y}^{-,t}$ given $\boldsymbol{y}^{t-1}$, the STERGM for $\boldsymbol{y}^t$ conditional on $\boldsymbol{y}^{t-1}$ is

$$P(\boldsymbol{y}^t|\boldsymbol{y}^{t-1};\boldsymbol{\theta}^t) = P(\boldsymbol{y}^{+,t}|\boldsymbol{y}^{t-1};\boldsymbol{\theta}^{+,t}) \times P(\boldsymbol{y}^{-,t}|\boldsymbol{y}^{t-1};\boldsymbol{\theta}^{-,t}) \tag{2.2}$$

with the respective formation and dissolution models:

$$\begin{aligned} P(\boldsymbol{y}^{+,t}|\boldsymbol{y}^{t-1};\boldsymbol{\theta}^{+,t}) &= \exp[\boldsymbol{\theta}^{+,t} \cdot \boldsymbol{g}^{+}(\boldsymbol{y}^{+,t}, \boldsymbol{y}^{t-1}) - \psi(\boldsymbol{\theta}^{+,t}, \boldsymbol{y}^{t-1})], \\ P(\boldsymbol{y}^{-,t}|\boldsymbol{y}^{t-1};\boldsymbol{\theta}^{-,t}) &= \exp[\boldsymbol{\theta}^{-,t} \cdot \boldsymbol{g}^{-}(\boldsymbol{y}^{-,t}, \boldsymbol{y}^{t-1}) - \psi(\boldsymbol{\theta}^{-,t}, \boldsymbol{y}^{t-1})]. \end{aligned}$$

The parameter $\boldsymbol{\theta}^t = (\boldsymbol{\theta}^{+,t}, \boldsymbol{\theta}^{-,t}) \in \mathbb{R}^p$ is a concatenation of $\boldsymbol{\theta}^{+,t} \in \mathbb{R}^{p_1}$ and $\boldsymbol{\theta}^{-,t} \in \mathbb{R}^{p_2}$ with $p_1 + p_2 = p$.

Notably, the normalizing constant in the formation model at a time point t:

$$\exp[\psi(\boldsymbol{\theta}^{+,t}, \boldsymbol{y}^{t-1})] = \sum_{\boldsymbol{y}^{+,t} \in \mathcal{Y}^{+,t}} \exp[\boldsymbol{\theta}^{+,t} \cdot \boldsymbol{g}^{+}(\boldsymbol{y}^{+,t}, \boldsymbol{y}^{t-1})]$$

is a sum over all possible networks in $\mathcal{Y}^{+,t}$, and that in the dissolution model is similar except for notational difference. Measuring these normalizing constants can be computationally intractable when the number of nodes n is large [HH06a]. Thus, for the change point detection problem described in Section 2.3, we adopt the pseudo-likelihood of an ERGM to estimate the parameters. For a network modeling problem, other parameter estimation methods exploit MCMC sampling [GT92, Kri17a] or Bayesian inference [CF11, TFC16] to circumvent the intractability of the normalizing constants.

In particular, we formulate the logarithm of the pseudo-likelihood of a time-heterogeneous STERGM $P(\boldsymbol{y}^T, \boldsymbol{y}^{T-1}, \ldots, \boldsymbol{y}^2 | \boldsymbol{y}^1; \boldsymbol{\theta}) = \prod_{t=2}^{T} P(\boldsymbol{y}^t | \boldsymbol{y}^{t-1}; \boldsymbol{\theta}^t)$ as

$$\begin{aligned} l(\boldsymbol{\theta}) = &\sum_{t=2}^{T} \sum_{(i,j) \in \mathbb{Y}} \left\{ \boldsymbol{y}_{ij}^{+,t} [\boldsymbol{\theta}^{+,t} \cdot \Delta \boldsymbol{g}_{ij}^{+}(\boldsymbol{y}^{+,t})] - \log \left\{ 1 + \exp[\boldsymbol{\theta}^{+,t} \cdot \Delta \boldsymbol{g}_{ij}^{+}(\boldsymbol{y}^{+,t})] \right\} \right\} + \\ &\sum_{t=2}^{T} \sum_{(i,j) \in \mathbb{Y}} \left\{ \boldsymbol{y}_{ij}^{-,t} [\boldsymbol{\theta}^{-,t} \cdot \Delta \boldsymbol{g}_{ij}^{-}(\boldsymbol{y}^{-,t})] - \log \left\{ 1 + \exp[\boldsymbol{\theta}^{-,t} \cdot \Delta \boldsymbol{g}_{ij}^{-}(\boldsymbol{y}^{-,t})] \right\} \right\} \end{aligned} \tag{2.3}$$

where $\boldsymbol{\theta} = (\boldsymbol{\theta}^2, \ldots, \boldsymbol{\theta}^T) \in \mathbb{R}^{\tau \times p}$ with $\tau = T - 1$. The change statistics $\Delta \boldsymbol{g}_{ij}^{+}(\boldsymbol{y}^{+,t})$ denote the change in $\boldsymbol{g}^{+}(\boldsymbol{y}^{+,t})$ when $\boldsymbol{y}_{ij}^{+,t}$ changes from 0 to 1, while rest of the $\boldsymbol{y}^{+,t}$ remains the same. The $\Delta \boldsymbol{g}_{ij}^{-}(\boldsymbol{y}^{-,t})$ is defined similarly. Since $\boldsymbol{y}^{+,t}$ and $\boldsymbol{y}^{-,t}$ inherit the dependence on $\boldsymbol{y}^{t-1}$ by construction, we use the implicit dynamic terms, $\boldsymbol{g}^{+}(\boldsymbol{y}^{+,t})$ with $\boldsymbol{g}^{+} : \mathcal{Y}^{+,t} \to \mathbb{R}^{p_1}$ and $\boldsymbol{g}^{-}(\boldsymbol{y}^{-,t})$ with $\boldsymbol{g}^{-} : \mathcal{Y}^{-,t} \to \mathbb{R}^{p_2}$, as discussed in [KH14].

The $l(\boldsymbol{\theta})$ in (2.3) is an approximation to the log-likelihood of (2.2) for $t = 2, \ldots, T$. We use the pseudo-likelihood for the optimization problem defined in Section 2.3 because it is computationally feasible comparing to using MCMC sampling or Bayesian inference. Furthermore, the number of rows in $\boldsymbol{\theta}$ is $\tau = T - 1$ instead of T due to the transition

probability $P(\boldsymbol{y}^t|\boldsymbol{y}^{t-1};\boldsymbol{\theta}^t)$ that is conditional on the previous network. Though $\boldsymbol{y}^t$ can be conditioned on more previous networks, we only discuss STERGM under the first-order Markov assumption in this chapter.

We now define the change points to be detected in terms of the parameters in STERGM. Let $\{B_k\}_{k=0}^{K+1} \subset \{1, 2, \ldots, T\}$ be a collection of ordered change points with $1 = B_0 < B_1 < \cdots < B_K < B_{K+1} = T$ such that

$$\boldsymbol{\theta}^{B_k} = \boldsymbol{\theta}^{B_k+1} = \cdots = \boldsymbol{\theta}^{B_{k+1}-1}, \quad k = 0, \ldots, K,$$

$$\boldsymbol{\theta}^{B_k} \neq \boldsymbol{\theta}^{B_{k+1}}, \quad k = 0, \ldots, K-1, \quad \text{and} \quad \boldsymbol{\theta}^{B_{K+1}} = \boldsymbol{\theta}^{B_K}.$$

Our goal is to recover the collection $\{B_k\}_{k=1}^{K}$ from a sequence of observed networks $\{\boldsymbol{y}^t\}_{t=1}^{T}$ with the number of change points K also unknown. Note that one or more components in $\boldsymbol{\theta}^{B_{k+1}} \in \mathbb{R}^p$ can be different from the parameter at the previous change point $\boldsymbol{\theta}^{B_k} \in \mathbb{R}^p$. For this setting, we present our method in the next section.

2.3 Group Fused Lasso for STERGM

2.3.1 Optimization Problem

Inspired by [VB10] and [BV11], we propose the following estimator for our change point detection problem:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \; -l(\boldsymbol{\theta}) + \lambda \sum_{i=1}^{\tau-1} \frac{\|\boldsymbol{\theta}_{i+1,\cdot} - \boldsymbol{\theta}_{i,\cdot}\|_2}{\boldsymbol{d}_i} \tag{2.4}$$

where $l(\boldsymbol{\theta})$ is formulated by (2.3). The Group Fused Lasso penalty expressed as the sum of Euclidean norms encourages sparsity of the parameter differences, while enforcing multiple components in $\boldsymbol{\theta}_{i+1,j} - \boldsymbol{\theta}_{i,j}$ across $j = 1, \ldots, p$ to change at the same group i. This is an effect that cannot be achieved with the ℓ_1 penalty of the differences. Along with the user-specified network statistics in the STERGM, the sequential parameter differences learned from the observed networks with (2.4) can reflect the magnitude of structural changes over time.

Furthermore, the term $\lambda > 0$ is a tuning parameter for the Group Fused Lasso penalty, and the term $\boldsymbol{d} \in \mathbb{R}^{\tau-1}$ is a position dependant weight [BV11] such that $\boldsymbol{d}_i = \sqrt{\tau/[i(\tau-i)]}$ for $i \in [1, \tau-1]$. Intuitively, the inverse of $\boldsymbol{d}_i$ assigns a greater weight to the time point that is far from the beginning and the end of a time span.

Figure 2.1 gives an overview of the proposed framework. The shaded circles on the top denote the sequence of observed networks as time passes from left to right. The dashed circles in the middle denote the sequences of formation networks $\boldsymbol{y}^{+,t}$ and dissolution networks $\boldsymbol{y}^{-,t}$ recovered from the observed networks. Note that each observed network is utilized multiple times to extract useful information that emphasizes the transition between consecutive time steps. We learn the parameters denoted by the dotted circles at the bottom, while monitoring the sequential parameter differences.

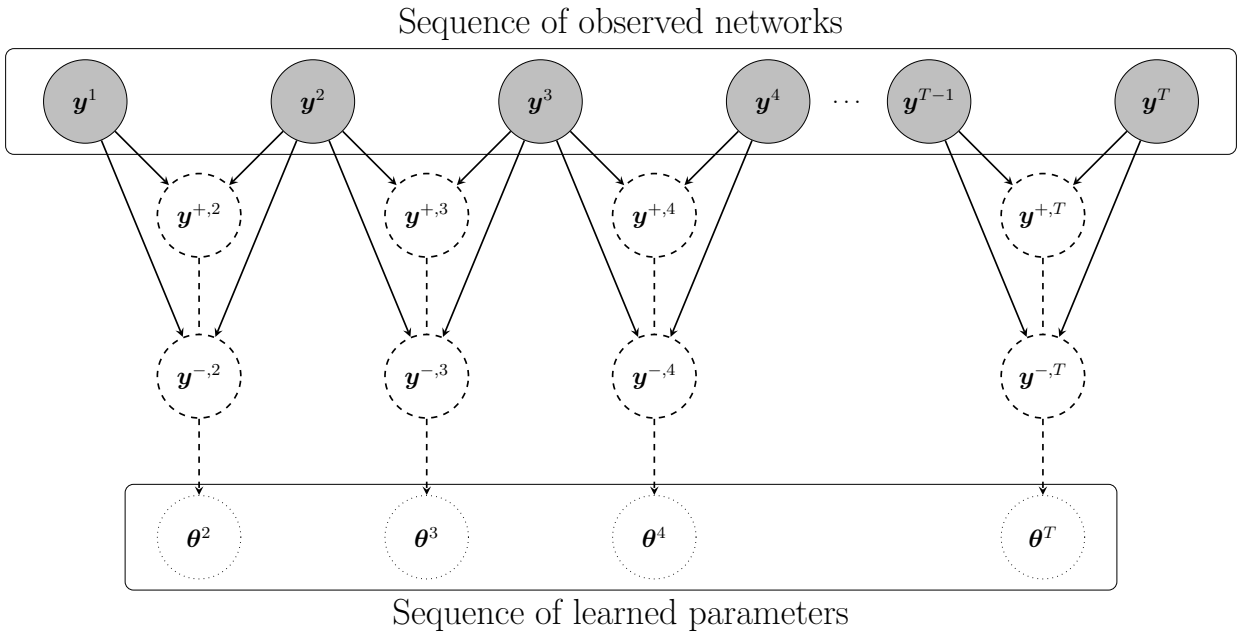

Figure 2.1: An illustration of change point model with STERGM.

To solve the problem in (2.4), we first introduce a slack variable $\boldsymbol{z} \in \mathbb{R}^{\tau \times p}$ and rewrite

the objective function as

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} -l(\boldsymbol{\theta}) + \lambda \sum_{i=1}^{\tau-1} \frac{\|\boldsymbol{z}_{i+1,\cdot} - \boldsymbol{z}_{i,\cdot}\|_2}{\boldsymbol{d}_i} \quad \text{subject to } \boldsymbol{\theta} = \boldsymbol{z}. \tag{2.5}$$

We then formulate the augmented Lagrangian as

$$\mathcal{L}_\alpha(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{\rho}) = -l(\boldsymbol{\theta}) + \lambda \sum_{i=1}^{\tau-1} \frac{\|\boldsymbol{z}_{i+1,\cdot} - \boldsymbol{z}_{i,\cdot}\|_2}{\boldsymbol{d}_i} + \mathrm{tr}[\boldsymbol{\rho}^\top(\boldsymbol{\theta} - \boldsymbol{z})] + \frac{\alpha}{2}\|\boldsymbol{\theta} - \boldsymbol{z}\|_F^2$$

where $\boldsymbol{\rho} \in \mathbb{R}^{\tau \times p}$ is the Lagrange multipliers and $\alpha \in \mathbb{R}$ is another penalty parameter for the augmentation term. Let $\boldsymbol{u} = \alpha^{-1}\boldsymbol{\rho}$ be the scaled dual variable, then

$$\mathcal{L}_\alpha(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{u}) = -l(\boldsymbol{\theta}) + \lambda \sum_{i=1}^{\tau-1} \frac{\|\boldsymbol{z}_{i+1,\cdot} - \boldsymbol{z}_{i,\cdot}\|_2}{\boldsymbol{d}_i} + \frac{\alpha}{2}\|\boldsymbol{\theta} - \boldsymbol{z} + \boldsymbol{u}\|_F^2 - \frac{\alpha}{2}\|\boldsymbol{u}\|_F^2. \tag{2.6}$$

[LH07] formulated a one-dimensional change point detection problem as a Lasso regression problem. Following [BV11], we make the change of variables $(\boldsymbol{\gamma}, \boldsymbol{\beta}) \in \mathbb{R}^{1\times p} \times \mathbb{R}^{(\tau-1)\times p}$ to formulate the augmented Lagrangian in (2.6) as a Group Lasso regression problem [YL06, ABD13], where

$$\boldsymbol{\gamma} = \boldsymbol{z}_{1,\cdot} \quad \text{and} \quad \boldsymbol{\beta}_{i,\cdot} = \frac{\boldsymbol{z}_{i+1,\cdot} - \boldsymbol{z}_{i,\cdot}}{\boldsymbol{d}_i} \quad \forall i \in [1, \tau - 1]. \tag{2.7}$$

Reversely, the matrix $\boldsymbol{z} \in \mathbb{R}^{\tau\times p}$ can also be collected by

$$\boldsymbol{z} = \mathbf{1}_{\tau,1}\boldsymbol{\gamma} + \boldsymbol{X}\boldsymbol{\beta}$$

where $\boldsymbol{X} \in \mathbb{R}^{\tau\times(\tau-1)}$ is a designed matrix with $\boldsymbol{X}_{i,j} = \boldsymbol{d}_j$ for $i > j$ and 0 otherwise. Plugging $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ into (2.6), we have

$$\mathcal{L}_\alpha(\boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\beta}, \boldsymbol{u}) = -l(\boldsymbol{\theta}) + \lambda \sum_{i=1}^{\tau-1} \|\boldsymbol{\beta}_{i,\cdot}\|_2 + \frac{\alpha}{2}\|\boldsymbol{\theta} - \mathbf{1}_{\tau,1}\boldsymbol{\gamma} - \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{u}\|_F^2 - \frac{\alpha}{2}\|\boldsymbol{u}\|_F^2.$$

Thus we derive an Alternating Direction Method of Multipliers (ADMM) to solve (2.5). The

resulting ADMM is given as:

$$\boldsymbol{\theta}^{(a+1)} = \arg\min_{\boldsymbol{\theta}} -l(\boldsymbol{\theta}) + \frac{\alpha}{2}\|\boldsymbol{\theta} - \boldsymbol{z}^{(a)} + \boldsymbol{u}^{(a)}\|_F^2, \tag{2.8}$$

$$\boldsymbol{\gamma}^{(a+1)}, \boldsymbol{\beta}^{(a+1)} = \arg\min_{\boldsymbol{\gamma},\boldsymbol{\beta}} \lambda \sum_{i=1}^{\tau-1} \|\boldsymbol{\beta}_{i,\cdot}\|_2 + \frac{\alpha}{2}\|\boldsymbol{\theta}^{(a+1)} - \mathbf{1}_{\tau,1}\boldsymbol{\gamma} - \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{u}^{(a)}\|_F^2, \tag{2.9}$$

$$\boldsymbol{u}^{(a+1)} = \boldsymbol{\theta}^{(a+1)} - \boldsymbol{z}^{(a+1)} + \boldsymbol{u}^{(a)}, \tag{2.10}$$

where a denotes the current ADMM iteration. Note that only in update (2.9) do we decompose $\boldsymbol{z}$ to work with $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ instead. Once the update (2.9) is completed within an ADMM iteration, we collect $\boldsymbol{z}^{(a+1)} = \mathbf{1}_{\tau,1}\boldsymbol{\gamma}^{(a+1)} + \boldsymbol{X}\boldsymbol{\beta}^{(a+1)}$ until the next decomposition of $\boldsymbol{z}$. We recursively implement the three updates until a convergence criterion is satisfied.

As in [BPC11], we also update the penalty parameter α to improve convergence and to reduce reliance on its initial choice. After the completion of an ADMM iteration, we calculate the respective primal and dual residuals:

$$r_{\text{primal}}^{(a)} = \sqrt{\frac{1}{\tau \times p}\sum_{i=1}^{\tau}\sum_{j=1}^{p}(\boldsymbol{\theta}_{ij}^{(a)} - \boldsymbol{z}_{ij}^{(a)})^2} \text{ and } r_{\text{dual}}^{(a)} = \sqrt{\frac{1}{\tau \times p}\sum_{i=1}^{\tau}\sum_{j=1}^{p}(\boldsymbol{z}_{ij}^{(a)} - \boldsymbol{z}_{ij}^{(a-1)})^2}$$

at the ath ADMM iteration. We update the penalty parameter α and the scaled dual variable $\boldsymbol{u}$ with the following schedule:

$$\alpha^{(a+1)} = 2\alpha^{(a)}, \boldsymbol{u}^{(a+1)} = \frac{1}{2}\boldsymbol{u}^{(a)} \text{ if } r_{\text{primal}}^{(a)} > 10 \times r_{\text{dual}}^{(a)},$$
$$\alpha^{(a+1)} = \frac{1}{2}\alpha^{(a)}, \boldsymbol{u}^{(a+1)} = 2\boldsymbol{u}^{(a)} \text{ if } r_{\text{dual}}^{(a)} > 10 \times r_{\text{primal}}^{(a)}.$$

Since STERGM is a probability distribution for the dynamic networks, in this work we stop ADMM learning until

$$\left|\frac{l(\boldsymbol{\theta}^{(a+1)}) - l(\boldsymbol{\theta}^{(a)})}{l(\boldsymbol{\theta}^{(a)})}\right| \leq \epsilon_{\text{tol}} \tag{2.11}$$

where ϵ_{tol} is a tolerance for the stopping criteria. Next, we discuss the updates (2.8) and (2.9) in detail.

2.3.2 Updating $\boldsymbol{\theta}$

In this section, we derive the Newton-Raphson method for learning $\boldsymbol{\theta}$ in the update (2.8). When the number of nodes n and the time steps T are large, updating $\boldsymbol{\theta}$ can be computationally expensive. To update $\boldsymbol{\theta}$ in a compact form, we first vectorize it as $\vec{\boldsymbol{\theta}} = \text{vec}_{\tau p}(\boldsymbol{\theta})$, and we construct

$$\boldsymbol{\Delta}^t = \begin{pmatrix} \boldsymbol{\Delta}^{+,t} & \\ & \boldsymbol{\Delta}^{-,t} \end{pmatrix} \text{ and } \boldsymbol{H} = \begin{pmatrix} \boldsymbol{\Delta}^2 & & \\ & \ddots & \\ & & \boldsymbol{\Delta}^T \end{pmatrix}.$$

The matrices $\boldsymbol{\Delta}^{+,t} \in \mathbb{R}^{E \times p_1}$ and $\boldsymbol{\Delta}^{-,t} \in \mathbb{R}^{E \times p_2}$ abbreviate the respective change statistics $\Delta \boldsymbol{g}^{+}_{ij}(\boldsymbol{y}^{+,t})$ and $\Delta \boldsymbol{g}^{-}_{ij}(\boldsymbol{y}^{-,t})$ that are ordered by the dyads. The matrix $\boldsymbol{H} \in \mathbb{R}^{2\tau E \times \tau p}$ that consists of the change statistics for $t = 2, \ldots, T$ is calculated before the implementation of ADMM.

We also calculate $\vec{\boldsymbol{\mu}} = h(\boldsymbol{H} \cdot \vec{\boldsymbol{\theta}}) \in \mathbb{R}^{2\tau E \times 1}$ where $h(x) = 1/(1 + \exp(-x))$ is the element-wise sigmoid function. To calculate the Hessian, we need to construct

$$\boldsymbol{W}^t = \begin{pmatrix} \boldsymbol{W}^{+,t} & \\ & \boldsymbol{W}^{-,t} \end{pmatrix} \text{ and } \boldsymbol{W} = \begin{pmatrix} \boldsymbol{W}^2 & & \\ & \ddots & \\ & & \boldsymbol{W}^T \end{pmatrix}$$

where $\boldsymbol{W}^{+,t} = \text{diag}(\boldsymbol{\mu}^{+,t}_{ij}(1 - \boldsymbol{\mu}^{+,t}_{ij})) \in \mathbb{R}^{E \times E}$ with $\boldsymbol{\mu}^{+,t}_{ij} = h(\boldsymbol{\theta}^{+,t} \cdot \Delta \boldsymbol{g}^{+}_{ij}(\boldsymbol{y}^{+,t}))$. The matrix $\boldsymbol{W}^{-,t} \in \mathbb{R}^{E \times E}$ is defined similarly except for notational difference.

Using the Newton-Raphson method, the $\vec{\boldsymbol{\theta}}$ can be updated iteratively by applying the following:

$$\vec{\boldsymbol{\theta}}_{c+1} = \vec{\boldsymbol{\theta}}_c - \big(\boldsymbol{H}^\top \boldsymbol{W} \boldsymbol{H} + \alpha \boldsymbol{I}_{\tau p}\big)^{-1} \cdot \big(- \boldsymbol{H}^\top (\vec{\boldsymbol{y}} - \vec{\boldsymbol{\mu}}) + \alpha (\vec{\boldsymbol{\theta}}_c - \vec{\boldsymbol{z}}^{(a)} + \vec{\boldsymbol{u}}^{(a)})\big) \tag{2.12}$$

where c denotes the current Newton-Raphson iteration. Both $\boldsymbol{W}$ and $\vec{\boldsymbol{\mu}}$ are also calculated based on $\vec{\boldsymbol{\theta}}_c$. The network data $\{\boldsymbol{y}^{+,t}, \boldsymbol{y}^{-,t}\}_{t=2}^{T}$ is vectorized in the form of $\vec{\boldsymbol{y}} \in \{0,1\}^{2\tau E \times 1}$ to align with the dyad order of the matrix $\boldsymbol{H} \in \mathbb{R}^{2\tau E \times \tau p}$. The derivations are provided in

the Appendix. Once the Newton-Raphson method is concluded within an ADMM iteration, we fold the updated vector $\vec{\boldsymbol{\theta}}$ back into a matrix as $\boldsymbol{\theta}^{(a+1)} = \text{vec}_{\tau,p}^{-1}(\vec{\boldsymbol{\theta}})$ before implementing the update in (2.9), which is discussed next.

2.3.3 Updating $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$

In this section, we derive the update in (2.9), which is equivalent to solving a Group Lasso problem. We decompose the matrix $\boldsymbol{z}$ to work with $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ instead. With ADMM, the updates on $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ do not require the network data and the change statistics, but the updates primarily rely on the $\boldsymbol{\theta}$ learned from the update (2.8).

By adapting the derivation from [VB10] and [BV11], the matrix $\boldsymbol{\beta} \in \mathbb{R}^{(\tau-1)\times p}$ can be updated in a block coordinate descent manner. Specifically, we iteratively apply the following equation to update $\boldsymbol{\beta}_{i,\cdot}$ for each block $i = 1, \ldots, \tau - 1$:

$$\boldsymbol{\beta}_{i,\cdot} \leftarrow \frac{1}{\alpha \boldsymbol{X}_{\cdot,i}^\top \boldsymbol{X}_{\cdot,i}} \Big(1 - \frac{\lambda}{\|\boldsymbol{s}_i\|_2}\Big)_+ \boldsymbol{s}_i \tag{2.13}$$

where $(\cdot)_+ = \max(\cdot, 0)$ and

$$\boldsymbol{s}_i = \alpha \boldsymbol{X}_{\cdot,i}^\top (\boldsymbol{\theta}^{(a+1)} + \boldsymbol{u}^{(a)} - \mathbf{1}_{\tau,1}\boldsymbol{\gamma} - \boldsymbol{X}_{\cdot,-i}\boldsymbol{\beta}_{-i,\cdot}).$$

The derivations are provided in the Appendix, and the convergence of the procedure is monitored by the Karush-Kuhn-Tucker (KKT) conditions: for all $\boldsymbol{\beta}_{i,\cdot} \neq \mathbf{0}$,

$$\lambda \frac{\boldsymbol{\beta}_{i,\cdot}}{\|\boldsymbol{\beta}_{i,\cdot}\|_2} - \alpha \boldsymbol{X}_{\cdot,i}^\top (\boldsymbol{\theta}^{(a+1)} + \boldsymbol{u}^{(a)} - \mathbf{1}_{\tau,1}\boldsymbol{\gamma} - \boldsymbol{X}\boldsymbol{\beta}) = \mathbf{0},$$

and for all $\boldsymbol{\beta}_{i,\cdot} = \mathbf{0}$,

$$\| -\alpha \boldsymbol{X}_{\cdot,i}^\top (\boldsymbol{\theta}^{(a+1)} + \boldsymbol{u}^{(a)} - \mathbf{1}_{\tau,1}\boldsymbol{\gamma} - \boldsymbol{X}\boldsymbol{\beta})\|_2 \leq \lambda.$$

Subsequently, for any $\boldsymbol{\beta} \in \mathbb{R}^{(\tau-1)\times p}$, the minimum in $\boldsymbol{\gamma} \in \mathbb{R}^{1\times p}$ is achieved at

$$\boldsymbol{\gamma} = (1/\tau)\mathbf{1}_{1,\tau} \cdot (\boldsymbol{\theta}^{(a+1)} + \boldsymbol{u}^{(a)} - \boldsymbol{X}\boldsymbol{\beta}).$$

Algorithm 1 Group Fused Lasso STERGM

Input: initialized parameters $\boldsymbol{\theta}^{(1)}$, $\boldsymbol{\gamma}^{(1)}$, $\boldsymbol{\beta}^{(1)}$, $\boldsymbol{u}^{(1)}$, tuning parameter λ, penalty parameter α, number of iterations for ADMM, Newton-Raphson, and Group Lasso A, C, D, vectorized network data $\vec{\boldsymbol{y}}$, network change statistics $\boldsymbol{H}$

for $a = 1, \cdots, A$ **do**

  $\vec{\boldsymbol{\theta}} = \text{vec}_{\tau p}(\boldsymbol{\theta}^{(a)})$, $\vec{\boldsymbol{z}}^{(a)} = \text{vec}_{\tau p}(\mathbf{1}_{\tau,1}\boldsymbol{\gamma}^{(a)} + X\boldsymbol{\beta}^{(a)})$, $\vec{\boldsymbol{u}}^{(a)} = \text{vec}_{\tau p}(\boldsymbol{u}^{(a)})$

  for $c = 1, \cdots, C$ **do**

    Let $\vec{\boldsymbol{\theta}}_{c+1}$ be updated according to (2.12)

  end for

  $\boldsymbol{\theta}^{(a+1)} = \text{vec}^{-1}_{\tau,p}(\vec{\boldsymbol{\theta}}_{c+1})$

  Set $\tilde{\boldsymbol{\gamma}} = \boldsymbol{\gamma}^{(a)}$ and $\tilde{\boldsymbol{\beta}} = \boldsymbol{\beta}^{(a)}$

  for $d = 1, \cdots, D$ **do**

    for $i = 1, \cdots, \tau - 1$ **do**

      Let $\tilde{\boldsymbol{\beta}}^{d+1}_{i,\cdot}$ be updated according to (2.13)

    end for

    $\tilde{\boldsymbol{\gamma}}^{d+1} = (1/\tau)\mathbf{1}_{1,\tau} \cdot (\boldsymbol{\theta}^{(a+1)} + \boldsymbol{u}^{(a)} - \boldsymbol{X}\tilde{\boldsymbol{\beta}}^{d+1})$

  end for

  $\boldsymbol{\gamma}^{(a+1)} = \tilde{\boldsymbol{\gamma}}^{d+1}$, $\boldsymbol{\beta}^{(a+1)} = \tilde{\boldsymbol{\beta}}^{d+1}$

  $\boldsymbol{z}^{(a+1)} = \mathbf{1}_{\tau,1}\boldsymbol{\gamma}^{(a+1)} + \boldsymbol{X}\boldsymbol{\beta}^{(a+1)}$

  $\boldsymbol{u}^{(a+1)} = \boldsymbol{\theta}^{(a+1)} - \boldsymbol{z}^{(a+1)} + \boldsymbol{u}^{(a)}$

end for

$\hat{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}^{(a+1)}$

Output: learned parameters $\hat{\boldsymbol{\theta}}$

Once the update (2.9) is concluded within an ADMM iteration, we collect $\boldsymbol{z} = \mathbf{1}_{\tau,1}\boldsymbol{\gamma} + \boldsymbol{X}\boldsymbol{\beta}$ and proceed to (2.10) to update the scaled dual variable $\boldsymbol{u} \in \mathbb{R}^{\tau \times p}$.

The algorithm to solve (2.5) via ADMM is presented in Algorithm 1. The complexity of an iteration for the Newton-Raphson method is $O(\tau^2 p^2)$ and that for the block coordinate descent method is $O(\tau(\tau-1)p)$. In general, the complexity of Algorithm 1 is at least of order $O(A[C\tau^2 p^2 + D\tau(\tau-1)p])$, where A, C, and D are the respective numbers of iterations for ADMM, Newton-Raphson, and Group Lasso. Next, we provide practical guidelines for our proposed method.

2.4 Change Point Localization and Model Selection

In this section, we discuss the choices of network statistics for change point detection with STERGM, followed by change point localization and model selection.

2.4.1 Network Statistics

As a distribution over dynamic networks, STERGM allows us to generate different networks that share similar structural patterns with the observed networks, by using a carefully designed MCMC sampling algorithm [Sni02]. Hence, in a dynamic network modeling problem with STERGM, network statistics are often chosen to signify the underlying process producing the observed networks or to capture important network effects interpreting for a research question.

In our change point detection problem with STERGM, network statistics are chosen to determine the types of structural changes that are searched for by the researchers. The R library `ergm` [HHB22] provides an extensive list of network statistics that boost the power of the proposed method. Since the underlying reasons that result in edge formation are usually different from those that result in edge dissolution, the choices of network statistics in the formation model can be different from those in the dissolution model. For an in-depth

discussion of network statistics in an ERGM framework, see [HRS03], [HH06a], [SPR06], [HGH08], [MHH08], [RPW09], and [BH22].

2.4.2 Data-driven Threshold

Intuitively, the location of a change point is the time step where the parameter of STERGM at time t differs from that at time $t-1$. To this end, we can calculate the parameter difference between consecutive time points in $\hat{\boldsymbol{\theta}} \in \mathbb{R}^{\tau \times p}$ as

$$\Delta\hat{\boldsymbol{\theta}}_i = \|\hat{\boldsymbol{\theta}}_{i+1,.} - \hat{\boldsymbol{\theta}}_{i,.}\|_2 \quad \forall i \in [1, \tau - 1]$$

and declare a change point when a parameter difference is greater than a threshold.

Though researchers can choose an arbitrary threshold for $\Delta\hat{\boldsymbol{\theta}}$ based on the sensitivity of the detection, in this work we provide a data-driven threshold with the following procedures. First we standardize the parameter differences $\Delta\hat{\boldsymbol{\theta}}$ as

$$\Delta\hat{\boldsymbol{\zeta}}_i = \frac{\Delta\hat{\boldsymbol{\theta}}_i - \text{median}(\Delta\hat{\boldsymbol{\theta}})}{\text{sd}(\Delta\hat{\boldsymbol{\theta}})} \quad \forall i \in [1, \tau - 1]. \tag{2.14}$$

Then the threshold based on the parameters learned from the data is constructed as

$$\epsilon_{\text{thr}} = \text{mean}(\Delta\hat{\boldsymbol{\zeta}}) + \mathcal{Z}_{1-\alpha} \times \text{sd}(\Delta\hat{\boldsymbol{\zeta}}) \tag{2.15}$$

where $\mathcal{Z}_{1-\alpha}$ is the $(1-\alpha)\%$ quantile of the standard Normal distribution. We declare a change point B_k when $\Delta\hat{\boldsymbol{\zeta}}_{B_k} > \epsilon_{\text{thr}}$. The data-driven threshold in (2.15) is intuitive, as the values $\Delta\hat{\boldsymbol{\zeta}}$ at the change points are greater than those in between the change points, derived from the Group Fused Lasso penalty. When tracing in a plot over time, the values $\Delta\hat{\boldsymbol{\zeta}}$ can exhibit the magnitude of structural changes, in terms of the network statistics specified in the STERGM.

By convention, we also implement two post-processing steps to finalize the detected change points $\{\hat{B}_k\}_{k=1}^{K}$. When the spacing between consecutive change points is less than a threshold or $\hat{B}_k - \hat{B}_{k-1} < \delta_{\text{spc}}$, we keep the detected change point with greater $\Delta\hat{\boldsymbol{\zeta}}$ value

to avoid clusters of nearby change points. Furthermore, as the endpoints of a time span are usually not of interest, we discard the change point $\hat{B}_k$ less than a threshold δ_{end} and greater than $T - \delta_{\text{end}}$. In Section 2.5, we set $\delta_{\text{spc}} = 5$, and we set $\delta_{\text{end}} = 5$ and $\delta_{\text{end}} = 10$ for the simulated and real data experiments, respectively.

2.4.3 Model Selection

Determining the optimal set of change points over multiple STERGMs learned with different tuning parameter λ, we can use Bayesian information criterion (BIC) to perform model selection. Considering an STERGM with learned $\hat{\boldsymbol{\theta}}$ and fixed λ, we have

$$\text{BIC}(\hat{\boldsymbol{\theta}}, \lambda) = -2l(\hat{\boldsymbol{\theta}}) + \log(TN_{\text{net}}) \times p \times \text{Seg}(\hat{\boldsymbol{\theta}}, \lambda). \tag{2.16}$$

For a list of λ, we choose the set of change points obtained from the STERGM with the lowest BIC value.

Different from the number of nodes n, the network size N_{net} is $\binom{n}{2}$ for an undirected network and $2 \times \binom{n}{2}$ for a directed network. In general, for a dyadic dependent network, the effective network size is often smaller than N_{net} and it may be difficult to quantify the effective size [HGH08]. In this work, we use $N_{\text{net}} = \binom{n}{2}$ and $N_{\text{net}} = 2 \times \binom{n}{2}$ for the respective undirected and directed networks to consider the greater network size, since the procedure is to select a model with the smallest BIC value. In a node clustering problem for a static network, [HRT07] also used the number of observed edges as N_{net} to quantify the effective network size. Furthermore, the term $\text{Seg}(\hat{\boldsymbol{\theta}}, \lambda)$ gives the number of segments between change points $\{\hat{B}_k\}_{k=0}^{K+1}$ that are learned with the λ. In other words, $\text{Seg}(\hat{\boldsymbol{\theta}}, \lambda) = K + 1$, where K is the number of detected change points.

2.5 Simulated and Real Data Experiments

In this section, we evaluate the proposed method on both simulated and real data. For simulated data, we use the following three metrics to compare the performance of the proposed and competing methods. The first metric is the absolute error $|\hat{K} - K|$ where $\hat{K}$ and K are the numbers of detected and true change points, respectively. The second metric is the one-sided Hausdorff distance defined as

$$d(\hat{\mathcal{C}}|\mathcal{C}) = \max_{c \in \mathcal{C}} \min_{\hat{c} \in \hat{\mathcal{C}}} |\hat{c} - c|,$$

where $\hat{\mathcal{C}}$ and $\mathcal{C}$ are the respective sets of detected and true change points. We also report the metric $d(\mathcal{C}|\hat{\mathcal{C}})$. When $\hat{\mathcal{C}} = \emptyset$, we define $d(\hat{\mathcal{C}}|\mathcal{C}) = \infty$ and $d(\mathcal{C}|\hat{\mathcal{C}}) = -\infty$. The third metric described in [BW20] is the coverage of a partition $\mathcal{G}$ by another partition $\mathcal{G}'$, defined as

$$C(\mathcal{G}, \mathcal{G}') = \frac{1}{T} \sum_{\mathcal{A} \in \mathcal{G}} |\mathcal{A}| \cdot \max_{\mathcal{A}' \in \mathcal{G}'} \frac{|\mathcal{A} \cap \mathcal{A}'|}{|\mathcal{A} \cup \mathcal{A}'|}$$

with $\mathcal{A}, \mathcal{A}' \subseteq [1, T]$. The $\mathcal{G}$ and $\mathcal{G}'$ are collections of intervals between consecutive change points for the respective true and detected change points. Throughout, the network statistics are calculated directly from the R library `ergm` [HHB22] and the formulations are provided in the Appendix.

2.5.1 Simulations

We simulate dynamic networks from two particular models to imitate realistic social patterns. We use the Stochastic Block Model to attain that participants with similar attributes tend to form communities, and we impose a time-dependent mechanism in the generation process. Also, we simulate dynamic networks from STERGM, which separately takes into account how relations form and dissolve over time, as their underlying social reasons are usually different.

For each specification, we provide 10 Monte Carlo simulations of dynamic networks. We let the time span $T = 100$ and the number of nodes $n = 50, 100, 500$. The true change

points are located at $t = 26, 51, 76$, so $K = 3$. The $K + 1 = 4$ intervals in the partition $\mathcal{G}$ are $\mathcal{A}_1 = [1, \ldots, 25]$, $\mathcal{A}_2 = [26, \ldots, 50]$, $\mathcal{A}_3 = [51, \ldots, 75]$, and $\mathcal{A}_4 = [76, \ldots, 100]$. In each specification, we report the means over 10 Monte Carlo trials for different evaluation metrics.

To detect the change points with our method, we initialize the penalty parameter $\alpha = 10$. We let the tuning parameter $\lambda = 10^b$ with $b \in \{-2, -1, \ldots, 6, 7\}$. For each λ, we run $A = 200$ iterations of ADMM and the stopping criterion in (2.11) uses $\epsilon_{\text{tol}} = 10^{-7}$. Within each ADMM iteration, we run $C = 20$ iterations of the Newton-Raphson method, and $D = 20$ iterations for Group Lasso. The stopping criteria for the Newton-Raphson method is $\|\vec{\boldsymbol{\theta}}_{c+1} - \vec{\boldsymbol{\theta}}_c\|_2 < 10^{-3}$. To construct the data-driven threshold in (2.15), we use the 90% quantile of the standard Normal distribution.

Two competitor methods, gSeg [CZ15] and kerSeg [SC22c] that are available in the respective R libraries `gSeg` [CZC20] and `kerSeg` [SC22b], are provided for comparison. We use networks (nets.) and network statistics (stats.) as two types of input data to the competing methods. For gSeg, we use the minimum spanning tree to construct the similarity graph, and we use the approximated p-value of the original edge-count scan statistic. The significance level is set to $\alpha = 0.05$. For kerSeg, we use the approximated p-value of the fGKCP$_1$ [SC22c] and we set the significance level $\alpha = 0.001$. Throughout, we remain on these settings, since they produce good performance on average for the competitors. Changing the above settings can improve their performance on some specifications, while severely jeopardizing their performance on other specifications.

Scenario 1: Stochastic Block Model (SBM)

As in [PYP22], we construct two probability matrices $\boldsymbol{P}, \boldsymbol{Q} \in [0, 1]^{n \times n}$ and they are defined as

$$\boldsymbol{P}_{ij} = \begin{cases} 0.5, & i, j \in \mathcal{B}_l,\ l \in [3], \\ 0.3, & \text{otherwise}, \end{cases} \quad \text{and} \quad \boldsymbol{Q}_{ij} = \begin{cases} 0.45, & i, j \in \mathcal{B}_l,\ l \in [3], \\ 0.2, & \text{otherwise}, \end{cases}$$

where $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ are evenly sized clusters that form a partition of $\{1, \ldots, n\}$. We then construct a sequence of matrices $\boldsymbol{E}^t$ for $t = 1, \ldots, T$ such that

$$\boldsymbol{E}_{ij}^t = \begin{cases} \boldsymbol{P}_{ij}, & t \in \mathcal{A}_1 \cup \mathcal{A}_3, \\ \boldsymbol{Q}_{ij}, & t \in \mathcal{A}_2 \cup \mathcal{A}_4. \end{cases}$$

Lastly, the networks are generated with $\rho \in \{0.0, 0.5, 0.9\}$ as a time-dependent mechanism. For any ρ and $t = 1, \ldots, T-1$, we let $\boldsymbol{y}_{ij}^1 \sim \text{Bernoulli}(\boldsymbol{E}_{ij}^1)$ and

$$\boldsymbol{y}_{ij}^{t+1} \sim \begin{cases} \text{Bernoulli}\big(\rho(1 - \boldsymbol{E}_{ij}^{t+1}) + \boldsymbol{E}_{ij}^{t+1}\big), & \boldsymbol{y}_{ij}^t = 1, \\ \text{Bernoulli}\big((1-\rho)\boldsymbol{E}_{ij}^{t+1}\big), & \boldsymbol{y}_{ij}^t = 0. \end{cases}$$

When $\rho = 0$, the probability to draw an edge for dyad (i, j) at time $t+1$ remains the same. This imposes a time-independent condition for a sequence of generated networks. On the contrary, when $\rho > 0$, the probability to draw an edge for dyad (i, j) becomes greater at time $t+1$ when there exists an edge at time t, and the probability becomes smaller when there does not exist an edge at time t.

Figure 2.2 exhibits examples of generated networks at particular time points. Visually, Scenario 1 produces adjacency matrices with block structures, and mutuality is an important pattern in these networks. To detect the change points with our method, we use two network statistics, edge count and mutuality, in both formation and dissolution models. In the competitor methods, besides the dynamic networks $\{\boldsymbol{y}^t\}_{t=1}^T$, we also use the edge count and mutuality in $\{\boldsymbol{g}(\boldsymbol{y}^t)\}_{t=1}^T$ as another specification. Tables 2.1, 2.2, and 2.3 display the means of evaluation metrics for different specifications.

As expected, the kerSeg method can achieve a good performance on the covering metric $C(\mathcal{G}, \mathcal{G}')$ when $\rho = 0$, since the time-independent setting aligns with the kerSeg's assumption. However, the performances of gSeg and kerSeg are worsened when $\rho > 0$. In particular, when the networks in the sequence are time-dependent, both gSeg and kerSeg methods can effectively detect the true change points, as the one-sided Hausdorff distance $d(\hat{\mathcal{C}}|\mathcal{C})$ are close

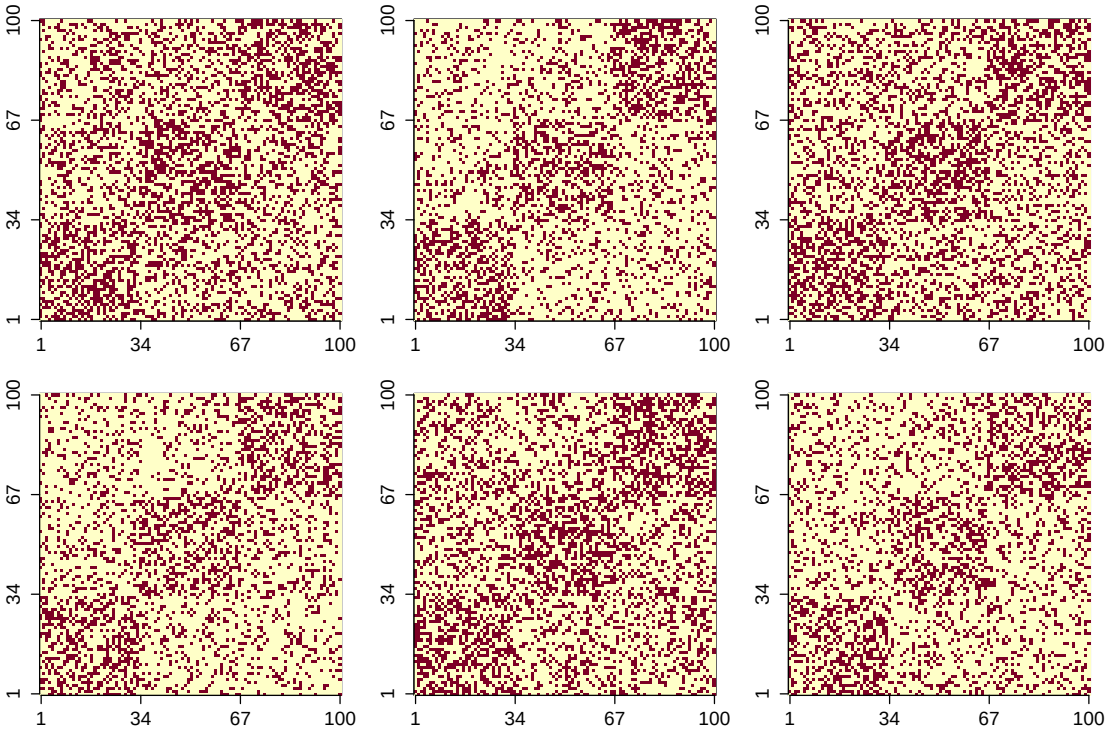

Figure 2.2: Examples of adjacency matrices generated from SBM with $\rho = 0.5$ and $n = 100$. In the first row, from left to right, each plot corresponds to the network at $t = 25, 50, 75$ respectively. In the second row, from left to right, each plot corresponds to the network at $t = 26, 51, 76$ respectively (the change points). In each display, a red dot indicates one and zero otherwise.

to zeros. Yet the reversed one-sided Hausdorff distance $d(\mathcal{C}|\hat{\mathcal{C}})$ and the absolute error $|\hat{K} - K|$ show that both gSeg and kerSeg tend to detect excessive number of change points as the sequences of networks become noisier under the time-dependent condition. Our CPDstergm method, on average, achieves smaller absolute error, smaller one-sided Hausdorff distances, and greater coverage of interval partitions, regardless of the temporal dependence.

Another aspect worth mentioning is the usage of the network statistics in the competitor methods. The performance of gSeg and kerSeg, in terms of the covering metric $C(\mathcal{G}, \mathcal{G}')$, improves significantly when we change the input data from networks to network statistics,

Table 2.1: Means of evaluation metrics for dynamic networks simulated from the Stochastic Block Model with $\rho = 0.0$.

ρ	n	Method	$\lvert\hat{K} - K\rvert$	$d(\hat{\mathcal{C}}\lvert\mathcal{C})$	$d(\mathcal{C}\lvert\hat{\mathcal{C}})$	$C(\mathcal{G}, \mathcal{G}')$
0.0	50	CPDstergm	0.3	0.8	2.2	95.35%
		gSeg (nets.)	2.9	inf	$-$inf	4.55%
		kerSeg (nets.)	**0**	**0**	**0**	**100**%
		gSeg (stats.)	2.1	inf	$-$inf	43.68%
		kerSeg (stats.)	0.1	**0**	0.3	99.70%
0.0	100	CPDstergm	1	0.8	5.8	89.07%
		gSeg (nets.)	2.9	inf	$-$inf	4.79%
		kerSeg (nets.)	**0**	**0**	**0**	**100**%
		gSeg (stats.)	1.9	inf	$-$inf	44.38%
		kerSeg (stats.)	**0**	**0**	**0**	**100**%
0.0	500	CPDstergm	**0**	1	1	97.07%
		gSeg (nets.)	3	inf	$-$inf	0%
		kerSeg (nets.)	**0**	**0**	**0**	**100**%
		gSeg (stats.)	2.1	inf	$-$inf	40.12%
		kerSeg (stats.)	**0**	**0**	**0**	**100**%

which demonstrates the potential of using network level summary statistics to represent the enormous amount of individual relations.

Scenario 2: Separable Temporal ERGM

In this scenario, we employ time-homogeneous STERGMs [KH14] between change points to generate sequences of dynamic networks, using the R package `tergm` [KH22]. For the following three specifications, we gradually increase the complexity of the network patterns, by adding more network statistics in the data generating process. First we use two network

Table 2.2: Means of evaluation metrics for dynamic networks simulated from the Stochastic Block Model with $\rho = 0.5$.

ρ	n	Method	$\lvert\hat{K}-K\rvert$	$d(\hat{\mathcal{C}}\lvert\mathcal{C})$	$d(\mathcal{C}\lvert\hat{\mathcal{C}})$	$C(\mathcal{G},\mathcal{G}')$
0.5	50	CPDstergm	**0.1**	1	**2.4**	**97.04**%
		gSeg (nets.)	12.9	**0**	19.4	27.20%
		kerSeg (nets.)	6.4	**0**	16.6	45.50%
		gSeg (stats.)	1.8	36.6	5.8	56.04%
		kerSeg (stats.)	0.7	**0**	4.4	94.60%
0.5	100	CPDstergm	**0**	1	**1**	**98.04**%
		gSeg (nets.)	12.3	**0**	19	27.80%
		kerSeg (nets.)	6	**0**	15.2	47.00%
		gSeg (stats.)	1.6	inf	$-$inf	53.50%
		kerSeg (stats.)	0.9	**0**	10	92.70%
0.5	500	CPDstergm	**0**	1	**1**	**98.04**%
		gSeg (nets.)	12.3	**0**	19.2	27.80%
		kerSeg (nets.)	4	**0**	12.7	52.20%
		gSeg (stats.)	1.7	36.6	3.9	58.59%
		kerSeg (stats.)	1.3	**0**	7.4	91.40%

statistics, edge count and mutuality, in both formation and dissolution models to let $p_{\text{sim}} = 4$. The parameters are

$$\boldsymbol{\theta}^{+,t}, \boldsymbol{\theta}^{-,t} = \begin{cases} -1,\ -2,\ -1,\ -2, & t \in \mathcal{A}_1 \cup \mathcal{A}_3, \\ -1,\ 1,\ -1,\ -1, & t \in \mathcal{A}_2 \cup \mathcal{A}_4. \end{cases}$$

Next, we include the number of triangles in both formation and dissolution models to let $p_{\text{sim}} = 6$. The parameters are

$$\boldsymbol{\theta}^{+,t}, \boldsymbol{\theta}^{-,t} = \begin{cases} -2,\ 2,\ -2,\ -1,\ 2,\ 1, & t \in \mathcal{A}_1 \cup \mathcal{A}_3, \\ -1.5,\ 1,\ -1,\ 2,\ 1,\ 1.5, & t \in \mathcal{A}_2 \cup \mathcal{A}_4. \end{cases}$$

Table 2.3: Means of evaluation metrics for dynamic networks simulated from the Stochastic Block Model with $\rho = 0.9$.

ρ	n	Method	$\lvert\hat{K} - K\rvert$	$d(\hat{\mathcal{C}}\lvert\mathcal{C})$	$d(\mathcal{C}\lvert\hat{\mathcal{C}})$	$C(\mathcal{G}, \mathcal{G}')$
0.9	50	CPDstergm	**0**	1	**1**	**98.04**%
		gSeg (nets.)	12.6	**0**	19.2	27.50%
		kerSeg (nets.)	11	**0**	18.7	32.00%
		gSeg (stats.)	6.7	5.4	16.9	58.45%
		kerSeg (stats.)	4.4	**0**	14	70.80%
0.9	100	CPDstergm	**0**	1	**1**	**98.04**%
		gSeg (nets.)	12.6	**0**	19	27.50%
		kerSeg (nets.)	12	**0**	19	28.00%
		gSeg (stats.)	5.6	1.6	18.8	62.66%
		kerSeg (stats.)	4	**0**	17.3	71.50%
0.9	500	CPDstergm	**0**	1	**1**	**98.04**%
		gSeg (nets.)	12.2	**0**	19	27.80%
		kerSeg (nets.)	12	**0**	19	28.00%
		gSeg (stats.)	7.4	0.2	19.1	58.96%
		kerSeg (stats.)	5.2	**0**	19	66.90%

Finally, we include the homophily for gender, an attribute assigned to each node, in both formation and dissolution models to let $p_{\text{sim}} = 8$. The parameters are

$$\boldsymbol{\theta}^{+,t}, \boldsymbol{\theta}^{-,t} = \begin{cases} -2,\ 2,\ -2,\ -1,\ -1,\ 2,\ 1,\ 1, & t \in \mathcal{A}_1 \cup \mathcal{A}_3, \\ -1.5,\ 1,\ -1,\ 1,\ 2,\ 1,\ 1.5,\ 2, & t \in \mathcal{A}_2 \cup \mathcal{A}_4. \end{cases}$$

The nodal attributes, $\boldsymbol{x}_i \in \{\text{Female}, \text{Male}\}$ for $i \in [n]$, are fixed across time t in the generation process.

Figure 2.3 exhibits examples of generated networks at particular time points. Specifically, Scenario 2 produces adjacency matrices that are sparse, which is often the case in reality. For

comparison, to detect the change points with our method, we use the network statistics that generate the networks in both formation and dissolution models. In the competitor methods, besides the dynamic networks, we also use the same network statistics that generate the networks as another specification. Tables 2.4, 2.5, and 2.6 display the means of evaluation metrics for different specifications.

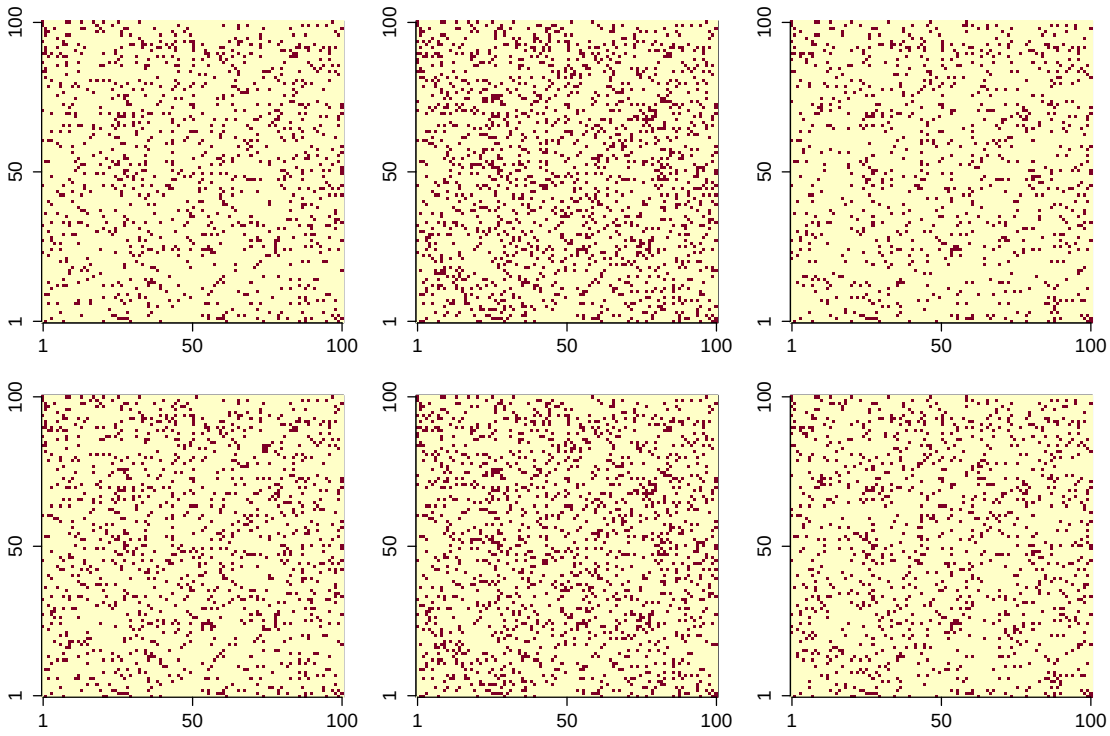

Figure 2.3: Examples of adjacency matrices generated from STERGM with $p_{\text{sim}} = 6$ and $n = 100$. In the first row, from left to right, each plot corresponds to the network at $t = 25, 50, 75$ respectively. In the second row, from left to right, each plot corresponds to the network at $t = 26, 51, 76$ respectively (the change points). In each display, a red dot indicates one and zero otherwise.

For $p_{\text{sim}} = 4$, the performance of the kerSeg method in terms of the covering metric $C(\mathcal{G}, \mathcal{G}')$ improves significantly when we change the input data from networks to network statistics. However, for $p_{\text{sim}} = 6$, both gSeg and kerSeg methods tend to detect excessive

Table 2.4: Means of evaluation metrics for dynamic networks simulated from the STERGM with $p_{\text{sim}} = 4$.

p_{sim}	n	Method	$\lvert\hat{K} - K\rvert$	$d(\hat{\mathcal{C}}\lvert\mathcal{C})$	$d(\mathcal{C}\lvert\hat{\mathcal{C}})$	$C(\mathcal{G}, \mathcal{G}')$
4	50	CPDstergm	**0**	0.1	0.1	99.80%
		gSeg (nets.)	1.9	21.7	12	48.83%
		kerSeg (nets.)	2.8	**0**	15.8	78.30%
		gSeg (stats.)	2.1	inf	$-$inf	43.61%
		kerSeg (stats.)	**0**	**0**	**0**	**100**%
4	100	CPDstergm	**0**	**0**	**0**	**100**%
		gSeg (nets.)	1.8	18.2	17.1	44.20%
		kerSeg (nets.)	2.6	**0**	15.6	77.40%
		gSeg (stats.)	2.1	inf	$-$inf	30.37%
		kerSeg (stats.)	0.1	**0**	0.2	99.80%
4	500	CPDstergm	**0**	1	**1**	**94.96**%
		gSeg (nets.)	12	**0**	19	28.00%
		kerSeg (nets.)	4.6	1.7	14.4	51.65%
		gSeg (stats.)	1.9	24.9	19.8	48.41%
		kerSeg (stats.)	4.3	1.4	19.4	74.02%

number of change points when the networks are highly dyadic dependent due to the inclusion of the triangle term. Using network statistics as input can no longer improve their performance. Our CPDstergm method, which dissects the network evolution using formation and dissolution models, can achieve a good result when the networks are both temporal and dyadic dependent. Lastly, for $p_{\text{sim}} = 8$, our method permits the inclusion of nodal attributes to facilitate the change detection, besides edge information. On average, our method produces smaller absolute error, smaller one-sided Hausdorff distances, and greater coverage of interval partitions.

Table 2.5: Means of evaluation metrics for dynamic networks simulated from the STERGM with $p_{\text{sim}} = 6$.

p_{sim}	n	Method	$\lvert\hat{K} - K\rvert$	$d(\hat{\mathcal{C}}\lvert\mathcal{C})$	$d(\mathcal{C}\lvert\hat{\mathcal{C}})$	$C(\mathcal{G}, \mathcal{G}')$
6	50	CPDstergm	**0.2**	1.6	**3**	**91.54**%
		gSeg (nets.)	12.3	**0**	19	27.90%
		kerSeg (nets.)	9.7	1.4	17.9	37.62%
		gSeg (stats.)	15.8	1.5	20.1	24.55%
		kerSeg (stats.)	9.4	3.9	18	35.86%
6	100	CPDstergm	**0**	1	**1**	**94.19**%
		gSeg (nets.)	12	**0**	19	28.00%
		kerSeg (nets.)	9.6	1	17.5	37.66%
		gSeg (stats.)	14.9	1.9	20.3	26.13%
		kerSeg (stats.)	8	5.4	16.7	38.45%
6	500	CPDstergm	**0**	1	**1**	**98.04**%
		gSeg (nets.)	12	**0**	19	28.00%
		kerSeg (nets.)	8.3	0.2	16.4	42.20%
		gSeg (stats.)	1.7	45.1	4.6	49.27%
		kerSeg (stats.)	6.1	3.1	15.3	55.24%

2.5.2 MIT Cellphone Data

The Massachusetts Institute of Technology (MIT) cellphone data [EP06] consists of human interactions via cellphone activity, among $n = 96$ participants for a duration of $T = 232$ days. The data were taken from 2004-09-15 to 2005-05-04 inclusive, which covers the winter and spring vacations in the MIT 2004-2005 academic calendar. For participant i and participant j, a connected edge $\boldsymbol{y}_{ij}^t = 1$ indicates that they had made at least one phone call on day t, and $\boldsymbol{y}_{ij}^t = 0$ indicates that they had made no phone call on day t.

As the data portrays human interactions, we use the number of (1) edges, (2) isolates,

Table 2.6: Means of evaluation metrics for dynamic networks simulated from the STERGM with $p_{\text{sim}} = 8$.

p_{sim}	n	Method	$\lvert\hat{K} - K\rvert$	$d(\hat{\mathcal{C}}\lvert\mathcal{C})$	$d(\mathcal{C}\lvert\hat{\mathcal{C}})$	$C(\mathcal{G}, \mathcal{G}')$
8	50	CPDstergm	**0.4**	1.7	**4.4**	**89.56**%
		gSeg (nets.)	13.3	**0**	19.6	27.20%
		kerSeg (nets.)	9.5	0.8	18.2	37.86%
		gSeg (stats.)	13.4	2.3	19.7	28.00%
		kerSeg (stats.)	8.7	4.8	18.3	36.51%
8	100	CPDstergm	**0**	1.6	**1.6**	**93.11**%
		gSeg (nets.)	12	**0**	19	28.00%
		kerSeg (nets.)	9.3	1.7	17.6	37.12%
		gSeg (stats.)	12.8	4.2	19.5	28.08%
		kerSeg (stats.)	8.2	5.8	18.6	36.55%
8	500	CPDstergm	**0.4**	12.3	**2.3**	**85.71**%
		gSeg (nets.)	12	**0**	19	28.00%
		kerSeg (nets.)	8.9	2	14.5	43.00%
		gSeg (stats.)	5.1	20.2	20.7	32.08%
		kerSeg (stats.)	9.6	2	17	37.95%

and (3) triangles to represent the occurrence of connections, the sparsity of social networks, and the transitive association of friendship, respectively. The three network statistics are used in both formation and dissolution models of our method. For the competitors, we use the three network statistics $\boldsymbol{g}(\boldsymbol{y}^t)$ as input data, since they provide better results than using the networks $\boldsymbol{y}^t$. Figure 2.4 displays $\Delta\hat{\boldsymbol{\zeta}}$ of Equation (2.14) and the detected change points of our method, as well as the results from the competitors. Moreover, Table 2.7 provides a list of potential nearby events that align with the detected change points of our method.

The two shaded areas in Figure 2.4 correspond to the winter and spring vacations, and

our method can punctually detect the pattern change in the contact behaviors. Both gSeg and kerSeg can also detect the beginning of the winter vacation, but their results on the spring vacation are slightly deviated. Furthermore, we detect a few spikes in the middle of October 2004, which correspond to the annual sponsor meeting that happened on 2004-10-21. About two-thirds of the participants have prepared and attended the annual sponsor meeting, and the majority of their time has contributed to achieve project goals throughout the week [EP06].

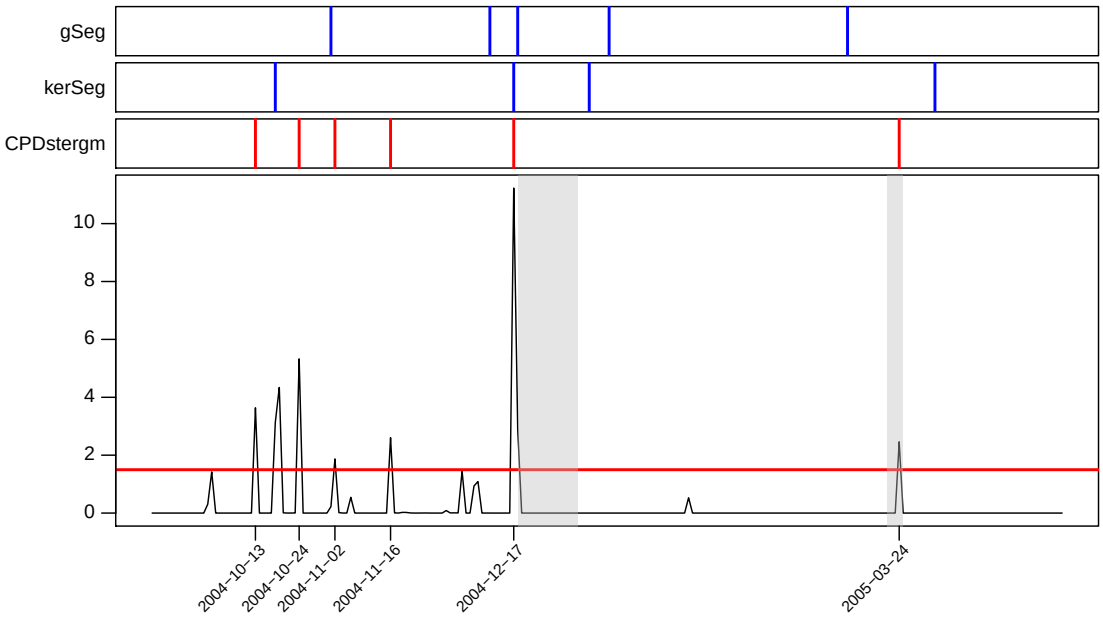

Figure 2.4: Visualization of $\Delta\hat{\boldsymbol{\zeta}}$ and the detected change points from our method for the MIT cellphone data. The detected change points from the competitors are also displayed. The two shaded areas correspond to the winter and spring vacations in the MIT 2004-2005 academic calendar. The data-driven threshold (red horizontal line) is calculated by (2.15) with $\mathcal{Z}_{0.9}$.

Table 2.7: Potential nearby events that align with the detected change points (CP) of our method

Detected CP	Potential nearby events
2004-10-13	Preparation for the Sponsor meeting
2004-10-24	2004-10-21 (Sponsor meeting)
2004-11-02	2004-11-02 (Presidential election)
2004-11-16	2004-11-17 (Last day to cancel subjects)
2004-12-17	2004-12-18 to 2005-01-02 (Winter vacation)
2005-03-24	2005-03-21 to 2005-03-25 (Spring vacation)

2.5.3 Stock Market Data

The stock market data consists of the weekly log returns of 29 stocks included in the Dow Jones Industrial Average (DJIA) index, and it is available in the R package `ecp` [JM15]. We consider the data from 2007-01-01 to 2010-01-04, which covers the 2008 worldwide economic crisis. We focus on the negative correlations among stock returns to detect the systematic anomalies in the financial market. Specifically, we first use a sliding window of width 4 to calculate the correlation matrices of the weekly log returns. We then truncate the correlation matrices by setting those entries which have negative values as 1, and the remaining as 0.

In the $T = 158$ networks, a connected edge $\boldsymbol{y}_{ij}^t = 1$ indicates the log returns of stock i and stock j are negative correlated over the four-week period that ends at week t. Moreover, the number of triangles can signify the volatility of the stock market, as the three stocks are mutually negative correlated. In general, the more triangles in a network, the more opposite movements among the stock returns, suggesting a large fluctuation in the market. On the contrary, when the number of triangles is small, the majority of the stock returns either increase or decrease at the same time, suggesting a stable trend in the market. To this end, we use the number of edges and triangles in both formation and dissolution models of our method. For the competitors, we use the networks $\boldsymbol{y}^t$ as input data, since they provide better

results than using the networks statistics $\boldsymbol{g}(\boldsymbol{y}^t)$. Figure 2.5 displays $\Delta\hat{\boldsymbol{\zeta}}$ of Equation (2.14) and the detected change points of our method, as well as the results from the competitors.

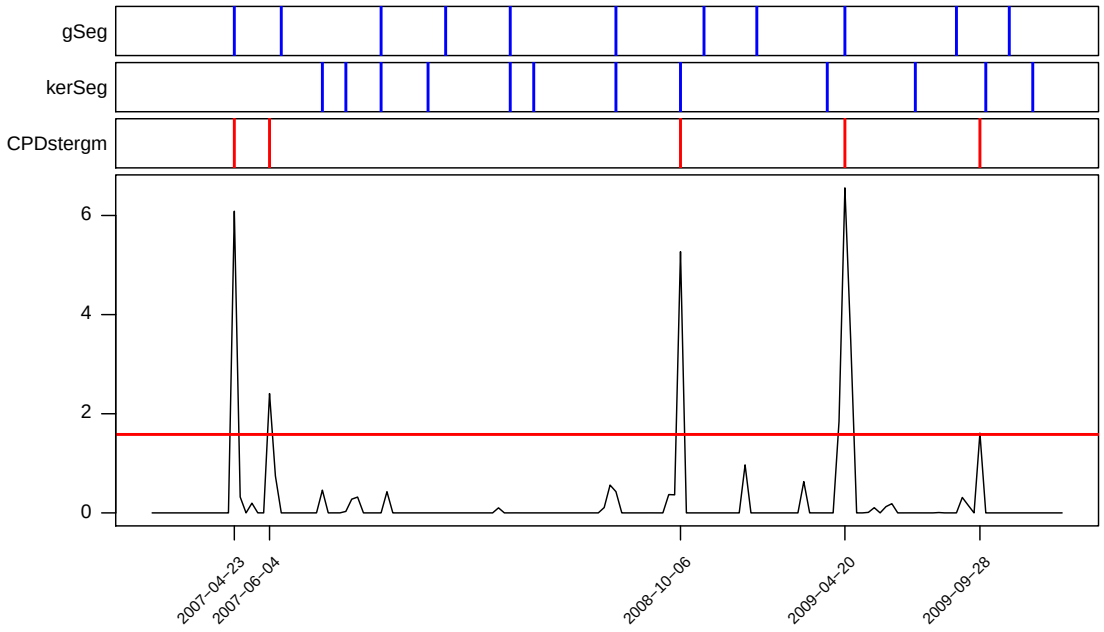

Figure 2.5: Visualization of $\Delta\hat{\boldsymbol{\zeta}}$ and the detected change points from our method for the stock market data. The detected change points from the competitors are also displayed. The data-driven threshold (red horizontal line) is calculated by (2.15) with $\mathcal{Z}_{0.9}$.

As expected, the stock market is volatile. The competitors have detected excessive number of change points, aligning with the smaller spikes in $\Delta\hat{\boldsymbol{\zeta}}$. Those change points can be detected by our method, if we manually lower the threshold to adjust the sensitivity. In this experiment, we focus on the top three spikes for real event interpretation. Table 2.8 presents the three detected change points and the potential nearby events. Given the networks are constructed using a sliding window, a detected change point indicates the pattern changes occur amid the four-week time span. As supporting evidence, the New Century Financial Corporation (NCFC) was the largest U.S. subprime mortgage lender in 2007, and the Lehman Brothers (LB) was one of the largest investment banks. Their bankruptcies caused

by the collapse of the mortgage industry severely fueled the worldwide financial crisis, which also led the DJIA to the bottom.

Table 2.8: Potential nearby events that align with the top three detected change points (CP) of our method

Detected CP	Potential nearby events
2007-04-23	2007-04-02 (NCFC filed for bankruptcy)
2008-10-06	2008-09-15 (LB filed for bankruptcy)
2009-04-20	2009-03-09 (DJIA bottomed)

2.6 Discussion

In this work, we study the change point detection problem in time series of graphs, which can serve as a prerequisite for dynamic network analysis. Essentially, we fit a time-heterogeneous STERGM while penalizing the sum of Euclidean norms of the parameter differences between consecutive time steps. The objective function with the Group Fused Lasso penalty is solved via Alternating Direction Method of Multipliers, and we adopt the pseudo-likelihood of STERGM to expedite the parameter estimation.

The STERGM [KH14] used in our method is a flexible model to fit dynamic networks with both dyadic and temporal dependence. It manages dyad formation and dissolution separately, as the underlying reasons that induce the two processes are usually different in reality. Furthermore, the ERGM suite [HHB22] provides an extensive list of network statistics to capture the structural changes, and we develop an R package `CPDstergm` to implement the proposed method.

Several improvements to our change point detection method are possible. Relational phenomena by nature have degrees of strength, and dichotomizing valued networks into binary networks may introduce biases for analysis [TB11]. We can extend the STERGM

with a valued ERGM [Kri12a, DC12a, CG20] to facilitate change point detection in dynamic valued networks. Moreover, the number of participants and their attributes are subject to change over time. It is necessary for a change point detection method to adjust the network sizes as in [KHM11], and to adapt the time-evolving nodal attributes by incorporating the Exponential-family Random Network Model (ERNM) as in [FH12a] and [FH13].

2.7 Appendix

2.7.1 Newton-Raphson Method for Updating $\boldsymbol{\theta}$

In this appendix, we derive the gradient and Hessian for the Newton-Raphson method to update $\boldsymbol{\theta}$. The first-order derivative of $l(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}^{+,t}$, the parameter in the formation model at a particular time point t, is

$$\begin{aligned}\nabla_{\boldsymbol{\theta}^{+,t}}\, l(\boldsymbol{\theta}) &= \sum_{ij}\Big\{\boldsymbol{y}_{ij}^{+,t}\Delta\boldsymbol{g}_{ij}^{+}(\boldsymbol{y}^{+,t}) - \frac{\exp[\boldsymbol{\theta}^{+,t}\cdot\Delta\boldsymbol{g}_{ij}^{+}(\boldsymbol{y}^{+,t})]}{1+\exp[\boldsymbol{\theta}^{+,t}\cdot\Delta\boldsymbol{g}_{ij}^{+}(\boldsymbol{y}^{+,t})]}\Delta\boldsymbol{g}_{ij}^{+}(\boldsymbol{y}^{+,t})\Big\}\\ &= \sum_{ij}(\boldsymbol{y}_{ij}^{+,t}-\boldsymbol{\mu}_{ij}^{+,t})[\Delta\boldsymbol{g}_{ij}^{+}(\boldsymbol{y}^{+,t})]\end{aligned}$$

where $\boldsymbol{\mu}_{ij}^{+,t} = h(\boldsymbol{\theta}^{+,t}\cdot\Delta\boldsymbol{g}_{ij}^{+}(\boldsymbol{y}^{+,t}))$. The $h(x) = 1/(1+\exp(-x))$ is the element-wise sigmoid function. Likewise, the first-order derivative of $l(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}^{-,t}$, the parameter in the dissolution model at a particular time point t, is similar except for notational difference.

Denote the objective function in (2.8) as $\mathcal{L}_\alpha(\boldsymbol{\theta})$. To update the parameters $\boldsymbol{\theta}\in\mathbb{R}^{\tau\times p}$ in a compact form, we first vectorize it as $\vec{\boldsymbol{\theta}} = \mathrm{vec}_{\tau p}(\boldsymbol{\theta})\in\mathbb{R}^{\tau p\times 1}$. The matrices $\boldsymbol{z}\in\mathbb{R}^{\tau\times p}$ and $\boldsymbol{u}\in\mathbb{R}^{\tau\times p}$ are also vectorized as $\vec{\boldsymbol{z}} = \mathrm{vec}_{\tau p}(\boldsymbol{z})\in\mathbb{R}^{\tau p\times 1}$ and $\vec{\boldsymbol{u}} = \mathrm{vec}_{\tau p}(\boldsymbol{u})\in\mathbb{R}^{\tau p\times 1}$. With the constructed matrices $\boldsymbol{H}\in\mathbb{R}^{2\tau E\times\tau p}$ and $\boldsymbol{W}\in\mathbb{R}^{2\tau E\times 2\tau E}$, the gradient of $\mathcal{L}_\alpha(\boldsymbol{\theta})$ with respect to $\vec{\boldsymbol{\theta}}$ is

$$\nabla_{\vec{\boldsymbol{\theta}}}\,\mathcal{L}_\alpha(\boldsymbol{\theta}) = -\boldsymbol{H}^\top(\vec{\boldsymbol{y}}-\vec{\boldsymbol{\mu}}) + \alpha\big(\vec{\boldsymbol{\theta}} - \vec{\boldsymbol{z}}^{(a)} + \vec{\boldsymbol{u}}^{(a)}\big)$$

where $\vec{\boldsymbol{\mu}} = h(\boldsymbol{H}\cdot\vec{\boldsymbol{\theta}})\in\mathbb{R}^{2\tau E\times 1}$.

Furthermore, the second order derivative of $l(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}^{+,t}$ is

$$\nabla^2_{\boldsymbol{\theta}^{+,t}} \, l(\boldsymbol{\theta}) = \sum_{ij} -\boldsymbol{\mu}_{ij}^{+,t}(1-\boldsymbol{\mu}_{ij}^{+,t})[\Delta \boldsymbol{g}_{ij}^{+}(\boldsymbol{y}^{+,t})\Delta \boldsymbol{g}_{ij}^{+}(\boldsymbol{y}^{+,t})^\top]$$

and the second order derivative of $l(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}^{-,t}$ is similar except for notational difference. Thus, the Hessian of $\mathcal{L}_\alpha(\boldsymbol{\theta})$ with respect to $\vec{\boldsymbol{\theta}} \in \mathbb{R}^{\tau p \times 1}$ is

$$\nabla^2_{\vec{\boldsymbol{\theta}}} \, \mathcal{L}_\alpha(\boldsymbol{\theta}) = \boldsymbol{H}^\top \boldsymbol{W} \boldsymbol{H} + \alpha \boldsymbol{I}_{\tau p}$$

where $\boldsymbol{I}_{\tau p}$ is the identity matrix. For fast implementation, it is possible to use the diagonal Hessian to approximate the above Hessian matrix. By using the Newton-Raphson method, the $\vec{\boldsymbol{\theta}} \in \mathbb{R}^{\tau p \times 1}$ is updated as

$$\vec{\boldsymbol{\theta}}_{c+1} = \vec{\boldsymbol{\theta}}_c - \big(\boldsymbol{H}^\top \boldsymbol{W} \boldsymbol{H} + \alpha \boldsymbol{I}_{\tau p}\big)^{-1} \cdot \big(- \boldsymbol{H}^\top (\vec{\boldsymbol{y}} - \vec{\boldsymbol{\mu}}) + \alpha (\vec{\boldsymbol{\theta}}_c - \vec{\boldsymbol{z}}^{(a)} + \vec{\boldsymbol{u}}^{(a)})\big)$$

where c denotes the current Newton-Raphson iteration. Note that both $\boldsymbol{W}$ and $\vec{\boldsymbol{\mu}}$ are also calculated based on $\vec{\boldsymbol{\theta}}_c$.

2.7.2 Group Lasso for Updating $\boldsymbol{\beta}$

In this appendix, we present the derivation of learning $\boldsymbol{\beta}$, which is equivalent to solving a Group Lasso problem. Denote the objective function in (2.9) as $\mathcal{L}_\alpha(\boldsymbol{\gamma}, \boldsymbol{\beta})$. When $\boldsymbol{\beta}_{i,\cdot} \neq \mathbf{0}$, the first-order derivative of $\mathcal{L}_\alpha(\boldsymbol{\gamma}, \boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}_{i,\cdot}$ is

$$\frac{\partial}{\partial \boldsymbol{\beta}_{i,\cdot}} \mathcal{L}_\alpha(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \lambda \frac{\boldsymbol{\beta}_{i,\cdot}}{\|\boldsymbol{\beta}_{i,\cdot}\|_2} - \alpha \boldsymbol{X}_{\cdot,i}^\top (\boldsymbol{\theta}^{(a+1)} + \boldsymbol{u}^{(a)} - \mathbf{1}_{\tau,1}\boldsymbol{\gamma} - \boldsymbol{X}_{\cdot,i}\boldsymbol{\beta}_{i,\cdot} - \boldsymbol{X}_{\cdot,-i}\boldsymbol{\beta}_{-i,\cdot})$$

where $\boldsymbol{X}_{\cdot,i} \in \mathbb{R}^{\tau \times 1}$ is the ith column of matrix $\boldsymbol{X} \in \mathbb{R}^{\tau \times (\tau-1)}$ and $\boldsymbol{\beta}_{i,\cdot} \in \mathbb{R}^{1 \times p}$ is the ith row of matrix $\boldsymbol{\beta} \in \mathbb{R}^{(\tau-1) \times p}$. Setting the gradient to $\mathbf{0}$, we have

$$\boldsymbol{\beta}_{i,\cdot} = (\alpha \boldsymbol{X}_{\cdot,i}^\top \boldsymbol{X}_{\cdot,i} + \frac{\lambda}{\|\boldsymbol{\beta}_{i,\cdot}\|_2})^{-1} \boldsymbol{s}_i \tag{2.17}$$

where

$$\boldsymbol{s}_i = \alpha \boldsymbol{X}_{\cdot,i}^\top (\boldsymbol{\theta}^{(a+1)} + \boldsymbol{u}^{(a)} - \mathbf{1}_{\tau,1}\boldsymbol{\gamma} - \boldsymbol{X}_{\cdot,-i}\boldsymbol{\beta}_{-i,\cdot}).$$

Taking the Euclidean norm of (2.17) on both sides and rearrange the terms, we have

$$\|\boldsymbol{\beta}_{i,\cdot}\|_2 = (\alpha \boldsymbol{X}_{\cdot,i}^\top \boldsymbol{X}_{\cdot,i})^{-1} (\|\boldsymbol{s}_i\|_2 - \lambda).$$

Plugging $\|\boldsymbol{\beta}_{i,\cdot}\|_2$ into (2.17), the solution of $\boldsymbol{\beta}_{i,\cdot}$ is

$$\boldsymbol{\beta}_{i,\cdot} = \frac{1}{\alpha \boldsymbol{X}_{\cdot,i}^\top \boldsymbol{X}_{\cdot,i}}(1 - \frac{\lambda}{\|\boldsymbol{s}_i\|_2})\boldsymbol{s}_i.$$

When $\boldsymbol{\beta}_{i,\cdot} = \mathbf{0}$, the subgradient $\boldsymbol{v}$ of $\|\boldsymbol{\beta}_{i,\cdot}\|_2$ needs to satisfy $\|\boldsymbol{v}\|_2 \leq 1$. Since

$$\mathbf{0} \in \lambda \boldsymbol{v} - \alpha \boldsymbol{X}_{\cdot,i}^\top (\boldsymbol{\theta}^{(a+1)} + \boldsymbol{u}^{(a)} - \mathbf{1}_{\tau,1}\boldsymbol{\gamma} - \boldsymbol{X}_{\cdot,-i}\boldsymbol{\beta}_{-i,\cdot}),$$

we obtain the condition that $\boldsymbol{\beta}_{i,\cdot}$ becomes $\mathbf{0}$ if $\|\boldsymbol{s}_i\|_2 \leq \lambda$. Therefore, we can iteratively apply the following equation to update $\boldsymbol{\beta}_{i,\cdot}$ for each $i = 1, \ldots, \tau - 1$:

$$\boldsymbol{\beta}_{i,\cdot} \leftarrow \frac{1}{\alpha \boldsymbol{X}_{\cdot,i}^\top \boldsymbol{X}_{\cdot,i}}\Big(1 - \frac{\lambda}{\|\boldsymbol{s}_i\|_2}\Big)_+ \boldsymbol{s}_i$$

where $(\cdot)_+ = \max(\cdot, 0)$. The matrix $\boldsymbol{X} \in \mathbb{R}^{\tau \times (\tau-1)}$ is constructed from the position dependent weight $\boldsymbol{d} \in \mathbb{R}^{\tau-1}$.

2.7.3 Network Statistics in Experiments

In this section, we provide the formulations of the network statistics used in the simulation and real data experiments. The network statistics of interest are chosen from the extensive list in `ergm` [HHB22], an R library for network analysis. Tables 2.9 and 2.10 display the formulations of network statistics used in the respective formation and dissolution models of our method for $t = 2, \ldots, T$. Moreover, Table 2.11 displays the formulations of network statistics used in the competitor methods for $t = 1, \ldots, T$. The formulations are referred to directed networks, and those for undirected networks are similar.

Table 2.9: Network statistics used in the formation model

Network Statistics	Formulation of $\boldsymbol{g}^{+}(\boldsymbol{y}^{+,t})$
Edge Count	$\sum_{ij} \boldsymbol{y}_{ij}^{+,t}$
Mutuality	$\sum_{i<j} \boldsymbol{y}_{ij}^{+,t} \boldsymbol{y}_{ji}^{+,t}$
Triangles	$\sum_{ijk} \boldsymbol{y}_{ij}^{+,t} \boldsymbol{y}_{jk}^{+,t} \boldsymbol{y}_{ik}^{+,t} + \sum_{ij<k} \boldsymbol{y}_{ij}^{+,t} \boldsymbol{y}_{jk}^{+,t} \boldsymbol{y}_{ki}^{+,t}$
Homophily	$\sum_{ij} \boldsymbol{y}_{ij}^{+,t} \times \mathbb{1}(\boldsymbol{x}_i = \boldsymbol{x}_j)$
Isolates	$\sum_i \mathbb{1}\big(\deg_{\text{in}}(\boldsymbol{y}^{+,t}, i) = 0 \wedge \deg_{\text{out}}(\boldsymbol{y}^{+,t}, i) = 0\big)$

Table 2.10: Network statistics used in the dissolution model

Network Statistics	Formulation of $\boldsymbol{g}^{-}(\boldsymbol{y}^{-,t})$
Edge Count	$\sum_{ij} \boldsymbol{y}_{ij}^{-,t}$
Mutuality	$\sum_{i<j} \boldsymbol{y}_{ij}^{-,t} \boldsymbol{y}_{ji}^{-,t}$
Triangles	$\sum_{ijk} \boldsymbol{y}_{ij}^{-,t} \boldsymbol{y}_{jk}^{-,t} \boldsymbol{y}_{ik}^{-,t} + \sum_{ij<k} \boldsymbol{y}_{ij}^{-,t} \boldsymbol{y}_{jk}^{-,t} \boldsymbol{y}_{ki}^{-,t}$
Homophily	$\sum_{ij} \boldsymbol{y}_{ij}^{-,t} \times \mathbb{1}(\boldsymbol{x}_i = \boldsymbol{x}_j)$
Isolates	$\sum_i \mathbb{1}\big(\deg_{\text{in}}(\boldsymbol{y}^{-,t}, i) = 0 \wedge \deg_{\text{out}}(\boldsymbol{y}^{-,t}, i) = 0\big)$

Table 2.11: Network statistics used in the competitor methods

Network Statistics	Formulation of $\boldsymbol{g}(\boldsymbol{y}^{t})$
Edge Count	$\sum_{ij} \boldsymbol{y}_{ij}^{t}$
Mutuality	$\sum_{i<j} \boldsymbol{y}_{ij}^{t} \boldsymbol{y}_{ji}^{t}$
Triangles	$\sum_{ijk} \boldsymbol{y}_{ij}^{t} \boldsymbol{y}_{jk}^{t} \boldsymbol{y}_{ik}^{t} + \sum_{ij<k} \boldsymbol{y}_{ij}^{t} \boldsymbol{y}_{jk}^{t} \boldsymbol{y}_{ki}^{t}$
Homophily	$\sum_{ij} \boldsymbol{y}_{ij}^{t} \times \mathbb{1}(\boldsymbol{x}_i = \boldsymbol{x}_j)$
Isolates	$\sum_i \mathbb{1}\big(\deg_{\text{in}}(\boldsymbol{y}^{t}, i) = 0 \wedge \deg_{\text{out}}(\boldsymbol{y}^{t}, i) = 0\big)$

CHAPTER 3

Change Point Detection in Dynamic Graphs with Generative Model

This chapter proposes a simple generative model to detect change points in time series of graphs. The proposed framework consists of learnable prior distributions for low-dimensional graph representations and of a decoder that can generate dynamic graphs from the latent representations. The informative prior distributions in the latent spaces are learned from observed data as empirical Bayes, and the expressive power of a generative model is exploited to assist change point detection. Specifically, the model parameters are learned via maximum approximate likelihood, with a Group Fused Lasso regularization. The optimization problem is then solved via Alternating Direction Method of Multipliers (ADMM), and Langevin Dynamics are recruited for posterior inference. Experiments in simulated and real data demonstrate the ability of the generative model in supporting change point detection with good performance.

3.1 Introduction

Networks are often used to represent relational phenomena in numerous domains [DLL21, HHL23, HBS23], and relational phenomena by nature progress in time. In recent decades, a plethora of models has been proposed to analyze the interaction between objects or people over time, including Temporal Exponential-family Random Graph Model [HFX10, KH14], Stochastic Actor-Oriented Model [Sni01, SBS10], and Relational Event Model [But08a,

BLS23]. Though these models incorporate the temporal aspect for analysis, network evolution is usually time-heterogeneous. Without taking the structural changes across dynamic networks into consideration, learning from the time series may not be meaningful. Hence, it is practical for researchers to localize the change points in time, before studying the evolving networks.

Various methodologies have been proposed to detect change points in dynamic networks. [CAA20] employed embedding methods to detect both anomalous graphs and anomalous vertices in time series of networks. [PS20] combined the multi-linear tensor regression model with a hidden Markov model, detecting changes based on the transition between the hidden states. [SKC23] learned a graph similarity function using a Siamese graph neural network to differentiate the graphs before and after a change point. Furthermore, [ZCL19] developed a screening algorithm that is based on an initial graphon estimation to detect change points. [HHR20] utilized the singular values of the Laplacian matrices as graph embedding to detect the differences across time. [CZ15], [CC19], and [SC22a] proposed a non-parametric approach to delineate the distributional differences over time, and [GA18] and [SC22c] exploited the patterns in high dimensions via a kernel-based method.

Inherently, network structures are complex due to highly dyadic dependency. Acquiring a low dimensional representation of the graph can summarize the enormous amount of individual relations to promote the downstream analysis. In particular, [SS22] and [KLC23] proposed to detect the structural changes using an Exponential-family Random Graph Model. Yet they relied on user-specified network statistics, which are usually not known to the modeler a priori. Moreover, [LBF21], [MBF22], and [GDZ23] developed different latent space models for dynamic graphs to detect changes, but they focused on node level representation, which may not be powerful enough to capture the information of the entire graph. On the other hand, generative models recently showed promising results in myriad applications, such as text generation with Large Language Model [DCL18, LLG19] and image generation with Diffusion Model [HJA20, RBL22]. Likewise, we are interested in exploring how generative

models can assist in the field of change point detection for dynamic graphs.

To tackle these challenges, we make the following contribution in this chapter:

- We learn graph level representations of network structures to facilitate change point detection. The informative prior distributions and a graph decoder are jointly learned via maximum approximate likelihood, with a multivariate total variation regularization.
- We derive an ADMM procedure to solve the resulting optimization problem. The prior distributions and the graph decoder are updated by inferring from the posterior distribution via Langevin Dynamics. Experiments show good performance of the generative model in supporting change point detection.

The rest of the chapter is organized as follows. Section 3.2 specifies the proposed framework. Section 3.3 presents the objective function with Group Fused Lasso regularization and the ADMM to solve the optimization problem. Section 3.4 discusses change points localization and model selection. Section 3.5 illustrates the proposed method on simulated and real data. Section 3.6 concludes the work with a discussion and potential future developments.

3.2 Generative Change Point Detection Model

3.2.1 Model Specification

For a node set $N = \{1, 2, \cdots, n\}$, we can use a network, graph, or adjacency matrix $\boldsymbol{y} \in \mathcal{Y}$ to represent the potential relations for all pairs $(i, j) \in \mathbb{Y} \subseteq N \times N$. The network $\boldsymbol{y}$ has dyad $\boldsymbol{y}_{ij} \in \{0, 1\}$ to indicate the absence or presence of a relation between node i and node j, thence $\mathcal{Y} \subseteq 2^{\mathbb{Y}}$. The relations in a network can be either directed or undirected. The undirected variant has $\boldsymbol{y}_{ij} = \boldsymbol{y}_{ji}$ for all (i, j).

Denote $\boldsymbol{y}^t \in \mathcal{Y}^t \subseteq 2^{\mathbb{Y}}$ as a network at a discrete time point t. The observed data is a sequence of networks $\boldsymbol{y}^1, \ldots, \boldsymbol{y}^T$. For each network $\boldsymbol{y}^t$, we assume there is a latent variable

$\boldsymbol{z}^t \in \mathbb{R}^d$ such that the network $\boldsymbol{y}^t$ can be generated from the latent variable with the following decoder:

$$\boldsymbol{y}^t \sim P(\boldsymbol{y}^t|\boldsymbol{z}^t) = \prod_{(i,j)\in\mathbb{Y}} \text{Bernoulli}(\boldsymbol{r}_{ij}(\boldsymbol{z}^t))$$

where $\boldsymbol{r}_{ij}(\boldsymbol{z}^t) = P(\boldsymbol{y}^t_{ij} = 1|\boldsymbol{z}^t)$ is the Bernoulli parameter for dyad (i, j) and it is elaborated in Section 3.2.2. Conditioning on the latent variable $\boldsymbol{z}^t \in \mathbb{R}^d$, we assume the network $\boldsymbol{y}^t \in \{0, 1\}^{n\times n}$ is dyadic independent.

We also impose a prior distribution to the latent variable as

$$\boldsymbol{z}^t \sim P(\boldsymbol{z}^t) = \mathcal{N}(\boldsymbol{\mu}^t, \boldsymbol{I}_d)$$

where $\boldsymbol{\mu}^t \in \mathbb{R}^d$ is a mean vector at time t and $\boldsymbol{I}_d$ is an identity matrix. Implicitly, $\boldsymbol{z}^t$ is considered as a graph level representation for $\boldsymbol{y}^t$. These representations are inferred from observed data, instead of explicitly specifying them. In this work, we learn the parameters $\{\boldsymbol{\mu}^t\}_{t=1}^T$ of the priors, facilitating change point detection in $\{\boldsymbol{y}^t\}_{t=1}^T$.

3.2.2 Simple Graph Decoder

The simple graph decoder $P(\boldsymbol{y}^t|\boldsymbol{z}^t) = \prod_{(i,j)\in\mathbb{Y}} P(\boldsymbol{y}^t_{ij}|\boldsymbol{z}^t)$ is formulated with a Bernoulli parameter for dyad (i, j) as

$$\boldsymbol{r}_{ij}(\boldsymbol{z}^t) = P(\boldsymbol{y}^t_{ij} = 1|\boldsymbol{z}^t) = \boldsymbol{g}_{ij}\big(\boldsymbol{h}(\boldsymbol{z}^t, t)\big).$$

The function $\boldsymbol{h}(\cdot)$ is parameterized by neural networks with $\boldsymbol{h} : \mathbb{R}^d \times \mathbb{N} \to \mathbb{R}^{n\times n}$. The function $\boldsymbol{g}(\cdot)$ is the element-wise sigmoid function with $\boldsymbol{g} : \mathbb{R}^{n\times n} \to [0, 1]^{n\times n}$.

In particular, we use multi-layer perceptrons (MLP), transferring the latent variable $\boldsymbol{z}^t \in \mathbb{R}^d$ to $\boldsymbol{U}^t \in \mathbb{R}^{n\times k}$ and $\boldsymbol{V}^t \in \mathbb{R}^{n\times k}$, respectively. We let the latent dimension d and k be smaller than the number of nodes n, and

$$\boldsymbol{h}(\boldsymbol{z}^t, t) = \begin{cases} \boldsymbol{U}^t{\boldsymbol{V}^t}^\top \in \mathbb{R}^{n\times n}, & \text{for directed network,} \\ \boldsymbol{U}^t{\boldsymbol{U}^t}^\top \in \mathbb{R}^{n\times n}, & \text{for undirected network.} \end{cases}$$

Figure 3.1 gives an overview of the proposed framework. Hierarchically, the graph level representation $\boldsymbol{z}^t$ progresses to node level representations $\boldsymbol{U}^t$ and $\boldsymbol{V}^t$ as an intermediate step, before the generation of network $\boldsymbol{y}^t$. The graph decoder $P_{\boldsymbol{\phi}}(\boldsymbol{y}^t|\boldsymbol{z}^t)$ with neural network parameter $\boldsymbol{\phi}$ is shared across $t = 1, \ldots, T$. It is worth pointing out the simplicity of our framework, without the need of encoders.

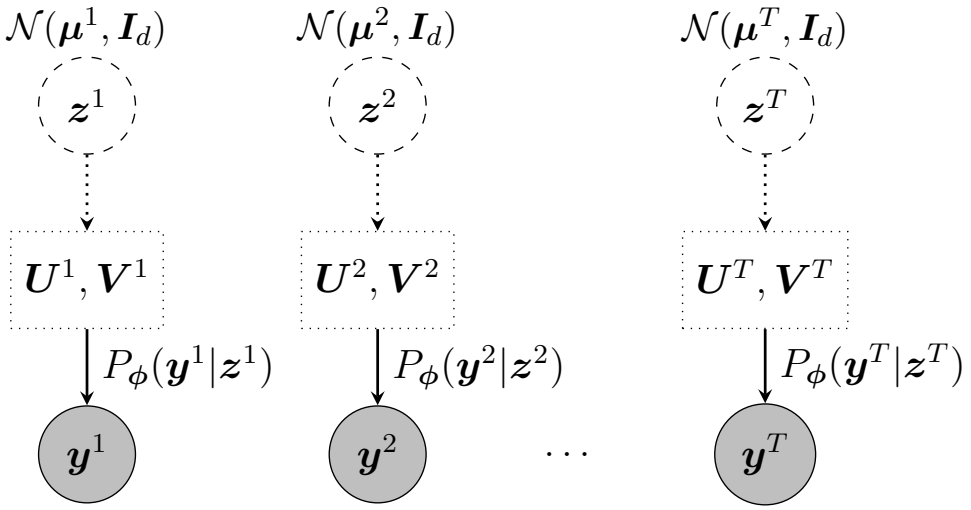

Figure 3.1: An overview of prior distributions and graph decoder.

3.2.3 Change Points

We now specify the change points to be detected in terms of the parameters of the prior distributions $P(\boldsymbol{z}^t)$ for $t = 1, \ldots, T$. Let $\{C_k\}_{k=0}^{K+1} \subset \{1, 2, \ldots, T\}$ be a collection of ordered change points with $1 = C_0 < C_1 < \cdots < C_K < C_{K+1} = T$ such that

$$\boldsymbol{\mu}^{C_k} = \boldsymbol{\mu}^{C_k+1} = \cdots = \boldsymbol{\mu}^{C_{k+1}-1}, \;\; k = 0, \ldots, K,$$

$$\boldsymbol{\mu}^{C_k} \neq \boldsymbol{\mu}^{C_{k+1}}, \;\; k = 0, \ldots, K-1, \;\; \text{and} \;\; \boldsymbol{\mu}^{C_{K+1}} = \boldsymbol{\mu}^{C_K}.$$

The change point detection problem comprises recovering the collection $\{C_k\}_{k=1}^{K}$ from a sequence of observed networks $\boldsymbol{y}^1, \ldots, \boldsymbol{y}^T$, where the number of change points K is also

unknown. For notational simplicity, we denote $\boldsymbol{\mu} \in \mathbb{R}^{T\times d}$ as a matrix where the t-th row corresponds to $\boldsymbol{\mu}^t \in \mathbb{R}^d$ for $t = 1, \ldots, T$.

3.3 Learning and Inference

3.3.1 Learning Priors from Dynamic Graphs

Inspired by [VB10] and [BV11], we formulate the change point detection problem as a Group Fused Lasso problem [ABD13] to infer the priors. Denote the log-likelihood of the model for $\boldsymbol{y}^1, \ldots, \boldsymbol{y}^T$ as $l(\boldsymbol{\phi}, \boldsymbol{\mu})$. We want to solve

$$\hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\mu}} = \arg\min_{\boldsymbol{\phi},\boldsymbol{\mu}} -l(\boldsymbol{\phi}, \boldsymbol{\mu}) + \lambda \sum_{t=1}^{T-1} \|\boldsymbol{\mu}^{t+1} - \boldsymbol{\mu}^t\|_2 \tag{3.1}$$

where $\lambda > 0$ is the tuning parameter for the penalty term. The Group Fused Lasso penalty encourages sparsity of the differences $\boldsymbol{\mu}^{t+1} - \boldsymbol{\mu}^t \in \mathbb{R}^d$, while allowing multiple sets of coordinates to change at the same time t, an effect that could not be possible with the ℓ_1 penalty of the differences.

To solve the optimization problem in (3.1), we first introduce a slack variable $\boldsymbol{\nu} \in \mathbb{R}^{T\times d}$ where $\boldsymbol{\nu}^t \in \mathbb{R}^d$ denotes the t-th row of $\boldsymbol{\nu}$, and we rewrite the original problem as

$$\begin{aligned} \hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\mu}} = &\arg\min_{\boldsymbol{\phi},\boldsymbol{\mu}} -l(\boldsymbol{\phi}, \boldsymbol{\mu}) + \lambda \sum_{t=1}^{T-1} \|\boldsymbol{\nu}^{t+1} - \boldsymbol{\nu}^t\|_2 \\ &\text{subject to } \ \boldsymbol{\mu} = \boldsymbol{\nu}. \end{aligned} \tag{3.2}$$

Then the augmented Lagrangian can be defined as

$$\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{\rho}) = -l(\boldsymbol{\phi}, \boldsymbol{\mu}) + \lambda \sum_{t=1}^{T-1} \|\boldsymbol{\nu}^{t+1} - \boldsymbol{\nu}^t\|_2 + \operatorname{tr}[\boldsymbol{\rho}^\top(\boldsymbol{\mu} - \boldsymbol{\nu})] + \frac{\kappa}{2}\|\boldsymbol{\mu} - \boldsymbol{\nu}\|_F^2$$

where $\boldsymbol{\rho} \in \mathbb{R}^{T\times d}$ is the Lagrange multipliers and $\kappa > 0$ is the penalty parameter for the augmentation term. Let $\boldsymbol{w} = \kappa^{-1}\boldsymbol{\rho} \in \mathbb{R}^{T\times d}$ be the scaled dual variable, the augmented Lagrangian can be updated to

$$\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{w}) = -l(\boldsymbol{\phi}, \boldsymbol{\mu}) + \lambda \sum_{t=1}^{T-1} \|\boldsymbol{\nu}^{t+1} - \boldsymbol{\nu}^t\|_2 + \frac{\kappa}{2}\|\boldsymbol{\mu} - \boldsymbol{\nu} + \boldsymbol{w}\|_F^2 - \frac{\kappa}{2}\|\boldsymbol{w}\|_F^2. \tag{3.3}$$

We further introduce two more variables $(\boldsymbol{\gamma}, \boldsymbol{\beta}) \in \mathbb{R}^{1\times d} \times \mathbb{R}^{(T-1)\times d}$ to ease the optimization. They are defined as

$$\boldsymbol{\gamma} = \boldsymbol{\nu}^1 \;\text{ and }\; \boldsymbol{\beta}_{t,\cdot} = \boldsymbol{\nu}^{t+1} - \boldsymbol{\nu}^t \;\; \forall\, t = 1, \ldots, T-1.$$

Reversely, the slack variable $\boldsymbol{\nu} \in \mathbb{R}^{T\times d}$ can be reconstructed as $\boldsymbol{\nu} = \mathbf{1}_{T,1}\boldsymbol{\gamma} + \boldsymbol{X}\boldsymbol{\beta}$, where $\boldsymbol{X}$ is a $T \times (T-1)$ matrix with entries $\boldsymbol{X}_{ij} = 1$ for $i > j$ and 0 otherwise. Substituting the $\boldsymbol{\nu}$ in (3.3) with $(\boldsymbol{\gamma}, \boldsymbol{\beta})$, we arrive at

$$\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\mu}, \boldsymbol{\gamma}, \boldsymbol{\beta}, \boldsymbol{w}) = -l(\boldsymbol{\phi}, \boldsymbol{\mu}) + \lambda \sum_{t=1}^{T-1} \|\boldsymbol{\beta}_{t,\cdot}\|_2 + \frac{\kappa}{2}\|\boldsymbol{\mu} - \mathbf{1}_{T,1}\boldsymbol{\gamma} - \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{w}\|_F^2 - \frac{\kappa}{2}\|\boldsymbol{w}\|_F^2.$$

Thus, we can derive the following Alternating Direction Method of Multipliers (ADMM) to solve (3.2):

$$\boldsymbol{\phi}_{(a+1)}, \boldsymbol{\mu}_{(a+1)} = \underset{\boldsymbol{\phi}, \boldsymbol{\mu}}{\arg\min}\;\; -l(\boldsymbol{\phi}, \boldsymbol{\mu}) + \frac{\kappa}{2}\|\boldsymbol{\mu} - \boldsymbol{\nu}_{(a)} + \boldsymbol{w}_{(a)}\|_F^2, \tag{3.4}$$

$$\boldsymbol{\gamma}_{(a+1)}, \boldsymbol{\beta}_{(a+1)} = \underset{\boldsymbol{\gamma}, \boldsymbol{\beta}}{\arg\min}\;\; \lambda \sum_{t=1}^{T-1} \|\boldsymbol{\beta}_{t,\cdot}\|_2 + \frac{\kappa}{2}\|\boldsymbol{\mu}_{(a+1)} - \mathbf{1}_{T,1}\boldsymbol{\gamma} - \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{w}_{(a)}\|_F^2, \tag{3.5}$$

$$\boldsymbol{w}_{(a+1)} = \boldsymbol{\mu}_{(a+1)} - \boldsymbol{\nu}_{(a+1)} + \boldsymbol{w}_{(a)}, \tag{3.6}$$

where a denotes the current ADMM iteration. We recursively implement the three updates until a convergence criterion is satisfied. Throughout, details about the implementation are provided in Appendix 3.7.2.

3.3.2 Parameters Update

3.3.2.1 Updating $\boldsymbol{\mu}$ and $\boldsymbol{\phi}$

Denote the objective function in (3.4) as $\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\mu})$. Setting the gradients of $\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\mu})$ with respect to the prior parameter $\boldsymbol{\mu}^t \in \mathbb{R}^d$ at a time point t to zeros, we have the following:

Proposition 1. *The solution for $\boldsymbol{\mu}^t$ at an iteration of our proposed ADMM algorithm is a weighted sum:*

$$\boldsymbol{\mu}^t = \frac{1}{1+\kappa}\mathbb{E}_{P(\boldsymbol{z}^t|\boldsymbol{y}^t)}(\boldsymbol{z}^t) + \frac{\kappa}{1+\kappa}(\boldsymbol{\nu}^t - \boldsymbol{w}^t) \tag{3.7}$$

between the conditional expectation of the latent variable under the posterior distribution $P(\boldsymbol{z}^t|\boldsymbol{y}^t)$ *and the difference between the slack and the scaled dual variables. The term* $\boldsymbol{w}^t \in \mathbb{R}^d$ *denotes the t-th row of the scaled dual variable* $\boldsymbol{w} \in \mathbb{R}^{T\times d}$*. The proof is provided in Appendix 3.7.1.1.*

Moreover, the gradient of $\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\mu})$ with respect to the graph decoder parameter $\boldsymbol{\phi}$ is calculated as

$$\nabla_{\boldsymbol{\phi}}\ \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\mu}) = -\sum_{t=1}^{T} \mathbb{E}_{P(\boldsymbol{z}^t|\boldsymbol{y}^t)}\Big(\nabla_{\boldsymbol{\phi}} \log P(\boldsymbol{y}^t|\boldsymbol{z}^t)\Big). \tag{3.8}$$

The parameter $\boldsymbol{\phi}$ can be updated efficiently through back-propagation.

Notably, calculating the solution in (3.7) and the gradient in (3.8) requires evaluating the conditional expectations under the posterior $P(\boldsymbol{z}^t|\boldsymbol{y}^t)$. We employ Langevin Dynamics to sample from the posterior, approximating the conditional expectations [XZN17, XLG18, NHH20, PHN20]. In particular, let subscript l be the time step of the Langevin Dynamics and s be a small step size. The Langevin Dynamics to draw samples from the posterior distribution $P(\boldsymbol{z}^t|\boldsymbol{y}^t)$ is achieved by iterating

$$\boldsymbol{z}_{l+1}^t = \boldsymbol{z}_l^t + s\big[\nabla_{\boldsymbol{z}^t} \log P(\boldsymbol{y}^t|\boldsymbol{z}^t) - (\boldsymbol{z}_l^t - \boldsymbol{\mu}^t)\big] + \sqrt{2s}\boldsymbol{\epsilon} \tag{3.9}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_d)$ is a random perturbation to the process. The derivation is provided in Appendix 3.7.1.2.

3.3.2.2 Updating $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$

In this section, we derive the update in (3.5), which is equivalent to solving a Group Lasso problem [YL06]. In particular, we decompose the matrix $\boldsymbol{\nu} \in \mathbb{R}^{T\times d}$ to work with $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$. With ADMM, the updates on $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ do not require the network data $\{\boldsymbol{y}^t\}_{t=1}^T$ but rely on the prior parameter $\boldsymbol{\mu} \in \mathbb{R}^{T\times d}$ learned from the update (3.4).

By adapting the derivation in [BV11], we have the following for our proposed ADMM algorithm:

Proposition 2. *The Group Lasso problem to update* $\boldsymbol{\beta} \in \mathbb{R}^{(T-1)\times d}$ *is solved in a block coordinate descent manner, by iteratively applying the following equation to each block* $t = 1, \ldots, T-1$*:*

$$\boldsymbol{\beta}_{t,\cdot} \leftarrow \frac{1}{\kappa \boldsymbol{X}_{\cdot,t}^\top \boldsymbol{X}_{\cdot,t}} \Big(1 - \frac{\lambda}{\|\boldsymbol{b}_t\|_2}\Big)_+ \boldsymbol{b}_t \tag{3.10}$$

where $(\cdot)_+ = \max(\cdot, 0)$ *and*

$$\boldsymbol{b}_t = \kappa \boldsymbol{X}_{\cdot,t}^\top (\boldsymbol{\mu}_{(a+1)} + \boldsymbol{w}_{(a)} - \mathbf{1}_{T,1}\boldsymbol{\gamma} - \boldsymbol{X}_{\cdot,-t}\boldsymbol{\beta}_{-t,\cdot}).$$

The proof is provided in Appendix 3.7.1.3.

The convergence of the procedure can be monitored by the Karush-Kuhn-Tucker (KKT) conditions: for all $\boldsymbol{\beta}_{t,\cdot} \neq \mathbf{0}$,

$$\lambda \frac{\boldsymbol{\beta}_{t,\cdot}}{\|\boldsymbol{\beta}_{t,\cdot}\|_2} - \kappa \boldsymbol{X}_{\cdot,t}^\top (\boldsymbol{\mu}_{(a+1)} + \boldsymbol{w}_{(a)} - \mathbf{1}_{T,1}\boldsymbol{\gamma} - \boldsymbol{X}\boldsymbol{\beta}) = \mathbf{0},$$

and for all $\boldsymbol{\beta}_{t,\cdot} = \mathbf{0}$,

$$\|-\kappa \boldsymbol{X}_{\cdot,t}^\top (\boldsymbol{\mu}_{(a+1)} + \boldsymbol{w}_{(a)} - \mathbf{1}_{T,1}\boldsymbol{\gamma} - \boldsymbol{X}\boldsymbol{\beta})\|_2 \leq \lambda.$$

Lastly, for any $\boldsymbol{\beta}$, the minimum in $\boldsymbol{\gamma} \in \mathbb{R}^{1\times d}$ is achieved at

$$\boldsymbol{\gamma} = (1/T)\mathbf{1}_{1,T} \cdot (\boldsymbol{\mu}_{(a+1)} + \boldsymbol{w}_{(a)} - \boldsymbol{X}\boldsymbol{\beta}).$$

In summary, the algorithm to solve the optimization problem in (3.2) via ADMM is presented in Algorithm 2. The steps to transform between $\boldsymbol{\nu}$ and $(\boldsymbol{\gamma}, \boldsymbol{\beta})$ within an ADMM iteration are omitted for succinctness.

3.4 Change Point Localization and Model Selection

3.4.1 Data-driven Threshold

To localize change points, we can calculate the differences between consecutive time points in $\hat{\boldsymbol{\mu}} \in \mathbb{R}^{T\times d}$ as

$$\Delta\hat{\boldsymbol{\mu}}^t = \|\boldsymbol{\mu}^t - \boldsymbol{\mu}^{t-1}\|_2 \quad \forall\, t \in [2, T]$$

Algorithm 2 Latent Space Group Fused Lasso

Input: learning iterations A, B, D, tuning parameter λ, penalty parameter κ, learning rates η, observed data $\{\boldsymbol{y}^t\}_{t=1}^T$, initialization $\{\boldsymbol{\phi}_{(1)}, \boldsymbol{\mu}_{(1)}, \boldsymbol{\gamma}_{(1)}, \boldsymbol{\beta}_{(1)}, \boldsymbol{w}_{(1)}\}$

for $a = 1, \cdots, A$ **do**

  for $t = 1, \cdots, T$ **do**

    draw m samples $\boldsymbol{z}_1^t, \ldots, \boldsymbol{z}_m^t$ from $P(\boldsymbol{z}^t|\boldsymbol{y}^t)$ according to Equation (3.9)

    $\boldsymbol{\mu}_{(a+1)}^t = \frac{1}{1+\kappa}(m^{-1}\sum_{u=1}^m \boldsymbol{z}_u^t) + \frac{\kappa}{1+\kappa}(\boldsymbol{\nu}^t - \boldsymbol{w}^t)$

  end for

  for $b = 1, \ldots, B$ **do**

    $\boldsymbol{\phi}_{(b+1)} = \boldsymbol{\phi}_{(b)} - \eta \times \nabla_{\boldsymbol{\phi}}\ \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\mu})$

  end for

  Set $\tilde{\boldsymbol{\gamma}}^{(1)} = \boldsymbol{\gamma}_{(a)}, \tilde{\boldsymbol{\beta}}^{(1)} = \boldsymbol{\beta}_{(a)}$

  for $d = 1, \ldots, D$ **do**

    for $t = 1, \ldots, T-1$ **do**

      Let $\tilde{\boldsymbol{\beta}}_{t,\cdot}^{(d+1)}$ be updated according to (3.10)

    end for

    $\tilde{\boldsymbol{\gamma}}^{(d+1)} = (1/T)\mathbf{1}_{1,T} \cdot (\boldsymbol{\mu}_{(a+1)} + \boldsymbol{w}_{(a)} - \boldsymbol{X}\tilde{\boldsymbol{\beta}}^{(d+1)})$

  end for

  Set $\boldsymbol{\gamma}_{(a+1)} = \tilde{\boldsymbol{\gamma}}^{(d+1)}, \boldsymbol{\beta}_{(a+1)} = \tilde{\boldsymbol{\beta}}^{(d+1)}$

  $\boldsymbol{w}_{(a+1)} = \boldsymbol{\mu}_{(a+1)} - \boldsymbol{\nu}_{(a+1)} + \boldsymbol{w}_{(a)}$

end for

$\hat{\boldsymbol{\mu}} \leftarrow \boldsymbol{\mu}_{(a+1)}$

Output: learned prior parameters $\hat{\boldsymbol{\mu}}$

and declare the change points when the corresponding differences are greater than a threshold. Though researchers can choose an arbitrary threshold based on their applications, we use the data-driven threshold constructed as

$$\epsilon_{\text{thr}} = \text{mean}(\Delta\hat{\boldsymbol{\zeta}}) + \mathcal{Z}_q \times \text{sd}(\Delta\hat{\boldsymbol{\zeta}}) \tag{3.11}$$

where $\mathcal{Z}_q$ is the $q\%$ quantile of the standard Normal distribution, and the standardized differences are defined as

$$\Delta\hat{\boldsymbol{\zeta}}^t = \frac{\Delta\hat{\boldsymbol{\mu}}^t - \text{median}(\Delta\hat{\boldsymbol{\mu}})}{\text{std}(\Delta\hat{\boldsymbol{\mu}})} \quad \forall\ t \in [2, T]. \tag{3.12}$$

Finally, we declare a change point C_k when $\Delta\hat{\boldsymbol{\zeta}}^{C_k} > \epsilon_{\text{thr}}$. The data-driven threshold in (3.11) is intuitive, as the $\Delta\hat{\boldsymbol{\zeta}}$ values close to the change points are greater than those far from the change points. Tracing in a plot over time, the $\Delta\hat{\boldsymbol{\zeta}}$ values can exhibit the magnitude of changes.

3.4.2 Model Selection

We use Cross-Validation to select λ. In particular, we split the original time series of graphs into training and testing sets: the training set consists of networks at odd indexed time points and the testing set consists of networks at even indexed time points. Fixed on a specific λ value, we learn the model parameters with the training set, and we evaluate the learned model with the testing set.

For a list of λ values, we choose the λ giving the maximal log-likelihood on the testing set. Note that the log-likelihood is approximated by Monte Carlo samples $\{\boldsymbol{z}_u^t\}_{u=1}^m$ drawn from the prior distribution $P(\boldsymbol{z}^t)$ as

$$\sum_{t=1}^{T} \log P(\boldsymbol{y}^t) \approx \sum_{t=1}^{T} \log \Big[\frac{1}{m}\sum_{u=1}^{m}[\prod_{(i,j)\in\mathbb{Y}} P(\boldsymbol{y}_{ij}^t|\boldsymbol{z}_u^t)]\Big].$$

Further computational details are discussed in Appendix 3.7.2. Anchored on the selected λ value, we learn the model parameters again with the full data, resulting the optimal set of detected change points.

3.5 Simulated and Real Data Experiments

In this section, we implement the proposed method on simulated and real data. For simulated data, we use the following metrics to evaluate the performance. The first metric is the absolute error $|\hat{K} - K|$, where $\hat{K}$ and K are the respective numbers of the detected and true change points. The second metric described in [MYW21] is the one-sided Hausdorff distance, which is defined as

$$d(\hat{\mathcal{C}}|\mathcal{C}) = \max_{c \in \mathcal{C}} \min_{\hat{c} \in \hat{\mathcal{C}}} |\hat{c} - c|$$

where $\hat{\mathcal{C}}$ and $\mathcal{C}$ are the respective sets of detected and true change points. We also report the reversed one-sided Hausdorff distance $d(\mathcal{C}|\hat{\mathcal{C}})$. By convention, when $\hat{\mathcal{C}} = \emptyset$, we let $d(\hat{\mathcal{C}}|\mathcal{C}) = \infty$ and $d(\mathcal{C}|\hat{\mathcal{C}}) = -\infty$. The last metric described in [BW20] is the coverage of a partition $\mathcal{G}$ by another partition $\mathcal{G}'$, which is defined as

$$C(\mathcal{G}, \mathcal{G}') = \frac{1}{T} \sum_{\mathcal{A} \in \mathcal{G}} |\mathcal{A}| \cdot \max_{\mathcal{A}' \in \mathcal{G}'} \frac{|\mathcal{A} \cap \mathcal{A}'|}{|\mathcal{A} \cup \mathcal{A}'|}$$

with $\mathcal{A}, \mathcal{A}' \subseteq [1, T]$. The $\mathcal{G}$ and $\mathcal{G}'$ are collections of intervals between consecutive change points for the respective true and detected change points.

3.5.1 Simulation Study

We simulate dynamic networks from three scenarios to compare the performance of the proposed and competing methods: Separable Temporal Exponential-family Random Graph Model (STERGM), Stochastic Block Model (SBM), and Recurrent Neural Network (RNN). For each scenario with different number of nodes $n = 50, 100$, we simulate 10 Monte Carlo trials of directed dynamic networks with time span $T = 100$. The true change points are located at $t = 26, 51, 76$ so $K = 3$. Moreover, the $K + 1 = 4$ intervals in partition $\mathcal{G}$ are $\mathcal{A}_1 = [1, \ldots, 25]$, $\mathcal{A}_2 = [26, \ldots, 50]$, $\mathcal{A}_3 = [51, \ldots, 75]$, and $\mathcal{A}_4 = [76, \ldots, 100]$. In each specification, we report the means and standard deviations over 10 Monte Carlo trials for the evaluation metrics. Throughout, the proposed method is named CPDlatent.

Three competitor methods, gSeg [CZ15], kerSeg [SC22c], and CPDstergm [KLC23], are provided for comparison. For CPDstergm, we first use two network statistics, edge count and mutuality, in both formation and dissolution models to let $p = 4$. We then add one more network statistic, number of triangles, in both formation and dissolution models to let $p = 6$ as another specification. For gSeg, we use the minimum spanning tree to construct the similarity graph, with the approximated p-value of the original edge-count scan statistic, and we set the significance level $\alpha = 0.05$. For kerSeg, we use the approximated p-value of the fGKCP$_1$, and we set the significance level $\alpha = 0.001$. Moreover, we use the networks (nets.) and the above three network statistics (stats.) as two types of input data to gSeg and kerSeg methods.

Scenario 1: Separable TERGM

In this scenario, we apply time-homogeneous STERGMs between change points to generate sequences of dynamic networks [KH14]. We use three network statistics, edge count, mutuality, and number of triangles, in both formation (F) and dissolution (D) models. The $p = 6$ parameters for each time point t are

$$\boldsymbol{\theta}_F^t, \boldsymbol{\theta}_D^t = \begin{cases} -2,\ 2,\ -2,\ -1,\ 2,\ 1, & t \in \mathcal{A}_1 \cup \mathcal{A}_3 \setminus 1, \\ -1.5,\ 1,\ -1,\ 2,\ 1,\ 1.5, & t \in \mathcal{A}_2 \cup \mathcal{A}_4. \end{cases}$$

Figure 3.2 exhibits examples of generated networks. Visually, STERGM produces adjacency matrices that are sparse, which is often the case in real world social networks.

Tables 3.1 displays the means and standard deviations of the evaluation metrics for comparison. Since the networks are directly sampled from STERGM, the CPDstergm method with correctly specified network statistics ($p = 6$) achieves the best result, in terms of greater converge of the intervals. However, when the network statistics are mis-specified ($p = 4$), the performance of CPDstergm is worsened, with greater gaps between the true and detected change points. Also, using either networks (nets.) or network statistics (stats.) cannot

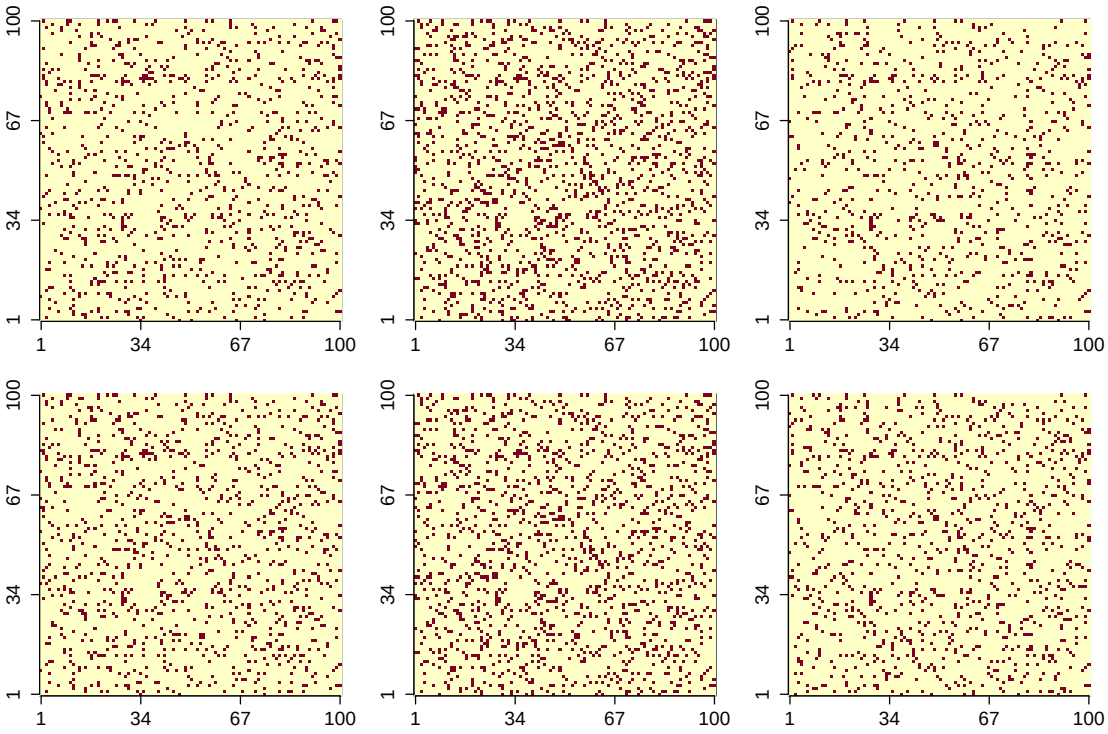

Figure 3.2: Examples of networks generated from STERGM with $n = 100$. In the first row, from left to right, each plot corresponds to the network at $t = 25, 50, 75$ respectively. In the second row, from left to right, each plot corresponds to the network at $t = 26, 51, 76$ respectively (the change points).

improve the performance of gSeg and kerSeg methods: the binary search approach tend to detect excessive number of change points. Our CPDlatent method, without the need of specifying network statistics, can achieve relatively good performance on average.

Scenario 2: Stochastic Block Model

In this scenario, we use Stochastic Block Model (SBM) to generate sequences of dynamic networks, and we impose a time-dependent mechanism in the generation process as in [MYP22].

Table 3.1: Means (stds.) of evaluation metrics for dynamic networks simulated from STERGM. The best coverage metric is bolded.

n	Method	$\lvert\hat{K}-K\rvert\downarrow$	$d(\hat{\mathcal{C}}\lvert\mathcal{C})\downarrow$	$d(\mathcal{C}\lvert\hat{\mathcal{C}})\downarrow$	$C(\mathcal{G},\mathcal{G}')\uparrow$
50	CPDlatent	0.1 (0.3)	4.3 (5.7)	2.6 (1.3)	90.87%
	CPDstergm$_{p=4}$	1.5 (0.8)	11.7 (7.5)	10.5 (2.3)	67.68%
	CPDstergm$_{p=6}$	0.2 (0.4)	1.6 (1.2)	3 (3.5)	**91.54**%
	gSeg (nets.)	12.3 (0.5)	0 (0)	19 (0)	27.90%
	kerSeg (nets.)	9.7 (0.9)	1.4 (0.9)	17.9 (1.2)	37.62%
	gSeg (stats.)	15.8 (0.7)	1.5 (0.5)	20.1 (0.3)	24.55%
	kerSeg (stats.)	9.4 (0.7)	3.9 (1.3)	18 (1.8)	35.86%
100	CPDlatent	0 (0)	3.9 (1.3)	3.9 (1.3)	91.33%
	CPDstergm$_{p=4}$	0.7 (0.6)	21.9 (10.3)	7.6 (4.3)	67.21%
	CPDstergm$_{p=6}$	0 (0)	1.1 (0.3)	1.1 (0.3)	**94.01**%
	gSeg (nets.)	12 (0)	0 (0)	19 (0)	28.00%
	kerSeg (nets.)	9.3 (0.8)	1 (0)	17.7 (0.6)	37.62%
	gSeg (stats.)	14.5 (2.3)	3.3 (3.6)	20.2 (0.4)	26.13%
	kerSeg (stats.)	8.5 (0.8)	4.5 (1.4)	17.3 (1.7)	36.92%

Two probability matrices $\boldsymbol{P}, \boldsymbol{Q} \in [0,1]^{n\times n}$ are constructed and they are defined as

$$\boldsymbol{P}_{ij} = \begin{cases} 0.5, & i,j \in \mathcal{B}_l,\ l \in [3], \\ 0.3, & \text{otherwise}, \end{cases} \quad \text{and} \quad \boldsymbol{Q}_{ij} = \begin{cases} 0.45, & i,j \in \mathcal{B}_l,\ l \in [3], \\ 0.2, & \text{otherwise}, \end{cases}$$

where $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$ are evenly sized clusters that form a partition of $\{1,\ldots,n\}$. Then a sequence of matrices $\boldsymbol{E}^t \in [0,1]^{n\times n}$ are arranged for $t = 1,\ldots,T$ such that

$$\boldsymbol{E}^t_{ij} = \begin{cases} \boldsymbol{P}_{ij}, & t \in \mathcal{A}_1 \cup \mathcal{A}_3, \\ \boldsymbol{Q}_{ij}, & t \in \mathcal{A}_2 \cup \mathcal{A}_4. \end{cases}$$

Lastly, the networks are generated with $\rho = 0.5$ as a time-dependent mechanism. For $t = 1, \ldots, T-1$, we let $\boldsymbol{y}_{ij}^1 \sim \text{Bernoulli}(\boldsymbol{E}_{ij}^1)$ and

$$\boldsymbol{y}_{ij}^{t+1} \sim \begin{cases} \text{Bernoulli}\big(\rho(1-\boldsymbol{E}_{ij}^{t+1}) + \boldsymbol{E}_{ij}^{t+1}\big), & \boldsymbol{y}_{ij}^t = 1, \\ \text{Bernoulli}\big((1-\rho)\boldsymbol{E}_{ij}^{t+1}\big), & \boldsymbol{y}_{ij}^t = 0. \end{cases}$$

With $\rho > 0$, the probability to form an edge for i, j becomes greater at time $t+1$ when there exists an edge at time t, and the probability becomes smaller when there does not exist an edge at time t. Figure 3.3 exhibits examples of generated networks. Visually, SBM produces adjacency matrices with block structures, where mutuality serves as an important pattern for the homophily within groups.

Tables 3.2 displays the means and standard deviations of the evaluation metrics for comparison. As expected, both CPDstergm methods with $p = 4$ and $p = 6$ that utilize the mutuality as a sufficient statistic for the detection can achieve good results, in terms of greater converge of the intervals. Furthermore, using network statistics (stats.) for both gSeg and kerSeg methods can improve their performance, comparing to using networks (nets.) as input data. Lastly, our CPDlatent method, which infers the features in latent space that induce the structural changes, achieves the best result for networks with block structures.

Scenario 3: Recurrent Neural Networks

In this scenario, we use Recurrent Neural Networks (RNN) to generate sequences of dynamic networks. Specifically, we sample latent variables from pre-defined priors, and we initialize the RNN with uniform weights. The graphs are then generated by the matrix multiplication defined in Section 3.2.2, using the output of RNN. The parameters for the pre-defined priors are

$$\boldsymbol{z}^t \sim \begin{cases} \mathcal{N}(-\mathbf{1},\ 0.1\boldsymbol{I}_d), & t \in \mathcal{A}_1 \cup \mathcal{A}_3, \\ \mathcal{N}(\mathbf{5},\ 0.1\boldsymbol{I}_d), & t \in \mathcal{A}_2 \cup \mathcal{A}_4. \end{cases}$$

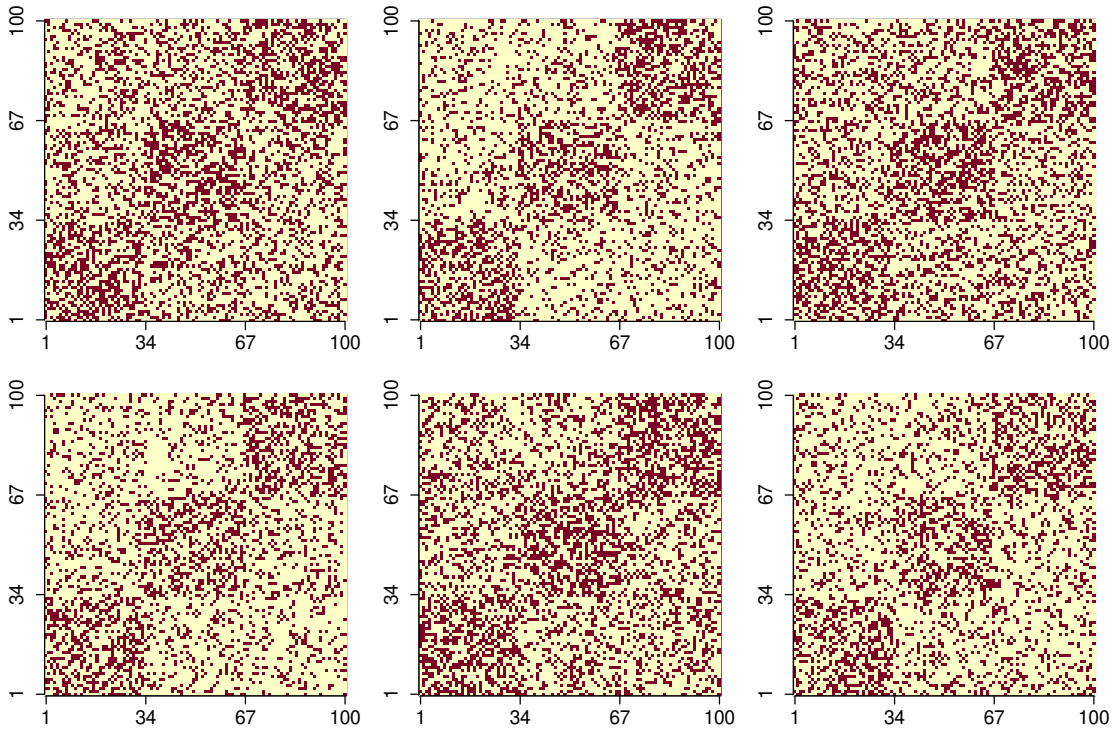

Figure 3.3: Examples of networks generated from SBM with $n = 100$. In the first row, from left to right, each plot corresponds to the network at $t = 25, 50, 75$ respectively. In the second row, from left to right, each plot corresponds to the network at $t = 26, 51, 76$ respectively (the change points).

Similar to the previous two scenarios, the simulation using RNN also imposes a time-dependent mechanism across dynamic networks. Figure 3.4 exhibits examples of generated networks. Visually, RNN produces adjacency matrices that are dense, and no discernible pattern can be noticed.

Tables 3.3 displays the means and standard deviations of the evaluation metrics for comparison. Because no structural pattern or suitable network statistics can be determined a priori, neither CPDstergm method with $p = 4$ nor with $p = 6$ can detect the change points accurately. Likewise, both gSeg and kerSeg methods that utilize the mis-specified network

Table 3.2: Means (stds.) of evaluation metrics for dynamic networks simulated from SBM. The best coverage metric is bolded.

n	Method	$\lvert\hat{K}-K\rvert\downarrow$	$d(\hat{\mathcal{C}}\lvert\mathcal{C})\downarrow$	$d(\mathcal{C}\lvert\hat{\mathcal{C}})\downarrow$	$C(\mathcal{G},\mathcal{G}')\uparrow$
50	CPDlatent	0 (0)	0.1 (0.3)	0.1 (0.3)	**99.80**%
	CPDstergm$_{p=4}$	0.1 (0.3)	1 (0)	2.4 (4.2)	97.04%
	CPDstergm$_{p=6}$	0.3 (0.5)	1 (0)	4.6 (5.6)	94.74%
	gSeg (nets.)	12.9 (1.8)	0 (0)	19.4 (0.8)	27.20%
	kerSeg (nets.)	6.4 (1.4)	0 (0)	16.6 (2.0)	45.50%
	gSeg (stats.)	2.2 (0.7)	inf (NA)	$-$inf (NA)	49.21%
	kerSeg (stats.)	0.9 (1.2)	0 (0)	5.6 (6.8)	93.50%
100	CPDlatent	0.1 (0.3)	0.1 (0.3)	1.3 (3.6)	**98.60**%
	CPDstergm$_{p=4}$	0 (0)	1 (0)	1 (0)	98.04%
	CPDstergm$_{p=6}$	0 (0)	1 (0)	1 (0)	98.04%
	gSeg (nets.)	12.3 (0.9)	0 (0)	19 (0)	27.80%
	kerSeg (nets.)	6 (0.8)	0 (0)	15.2 (2.0)	47.00%
	gSeg (stats.)	2 (0.4)	inf (NA)	$-$inf (NA)	55.75%
	kerSeg (stats.)	0.9 (0.7)	0 (0)	9.6 (7.6)	93.40%

statistics (stats.) cannot produce satisfactory performance. Notably, the kerSeg method that exploits the features in high dimension with networks (nets.) instead of user-specified network statistics (stats.) can deliver a good result. Lastly, our CPDlatent method that first infers the graph level representations from the complex network structures and then utilize them to detect the change points yields the best result.

3.5.2 MIT Cellphone Data

The Massachusetts Institute of Technology (MIT) cellphone data [EP06] depicts human interactions via phone call activities among $n = 96$ participants spanning $T = 232$ days.

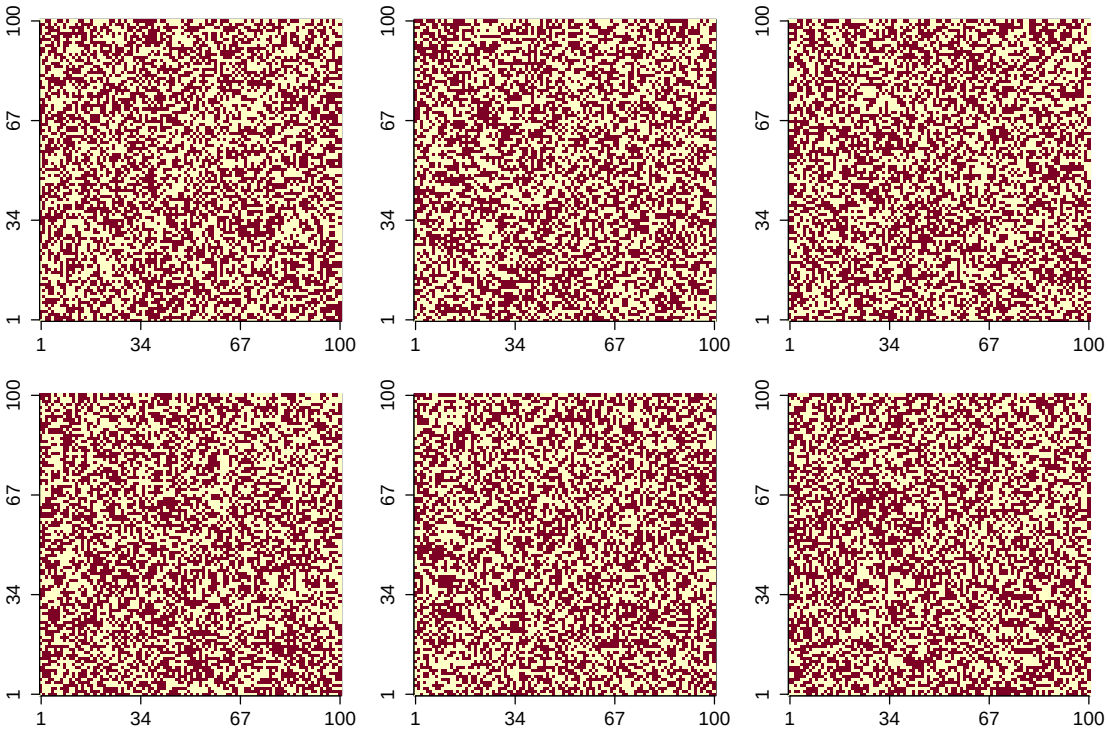

Figure 3.4: Examples of networks generated from RNN with $n = 100$. In the first row, from left to right, each plot corresponds to the network at $t = 25, 50, 75$ respectively. In the second row, from left to right, each plot corresponds to the network at $t = 26, 51, 76$ respectively (the change points).

An edge $\boldsymbol{y}_{ij}^t = 1$ in the constructed networks indicates that participant i and participant j had made phone calls on day t, and $\boldsymbol{y}_{ij}^t = 0$ otherwise. The data ranges from 2004-09-15 to 2005-05-04, covering the winter break in the MIT academic calendar.

We detect the change points with our proposed CPDlatent method, and we use network statistics as input data to the three competitor methods. Specifically, we use the number of edges, isolates, and triangles to capture the frequency of connections, the sparsity of social interaction, and the transitive association among friends, respectively. Figure 3.5 displays $\Delta\hat{\boldsymbol{\zeta}}$ of Equation (3.12), and the detected change points from our method and competitor methods.

Table 3.3: Means (stds.) of evaluation metrics for dynamic networks simulated from RNN. The best coverage metric is bolded.

n	Method	$\lvert\hat{K}-K\rvert\downarrow$	$d(\hat{\mathcal{C}}\lvert\mathcal{C})\downarrow$	$d(\mathcal{C}\lvert\hat{\mathcal{C}})\downarrow$	$C(\mathcal{G},\mathcal{G}')\uparrow$
50	CPDlatent	0 (0)	1.8 (0.7)	1.8 (0.7)	**94.77**%
	CPDstergm$_{p=4}$	2.0 (1.7)	6.0 (7.7)	15.2 (4.9)	72.10%
	CPDstergm$_{p=6}$	1.0 (0.4)	18.5 (9.4)	14.3 (2.9)	60.25%
	gSeg (nets.)	2.3 (0.6)	inf (NA)	−inf (NA)	29.42%
	kerSeg (nets.)	1.5 (0.9)	1.4 (0.7)	5.3 (3.3)	89.25%
	gSeg (stats.)	2.9 (0.3)	inf (NA)	−inf (NA)	2.47%
	kerSeg (stats.)	2.8 (0.4)	inf (NA)	−inf (NA)	9.89%
100	CPDlatent	0 (0)	2.5 (0.7)	2.5 (0.7)	**91.96**%
	CPDstergm$_{p=4}$	2.0 (1.4)	10.6 (8.0)	14.1 (3.1)	60.37%
	CPDstergm$_{p=6}$	1.2 (1.3)	20.6 (12.6)	15.2 (5.9)	53.21%
	gSeg (nets.)	3 (0)	inf (NA)	−inf (NA)	0%
	kerSeg (nets.)	1.4 (0.7)	1.9 (0.7)	5.4 (1.9)	88.95%
	gSeg (stats.)	2.9 (0.3)	inf (NA)	−inf (NA)	4.27%
	kerSeg (stats.)	3 (0)	inf (NA)	−inf (NA)	0%

Furthermore, Table 3.4 provides a list of potential events, aligning with the detected change points from our method.

Without specifying the structural changes to search for, our method can punctually detect the beginning of the winter break, which is the major event that alters the interaction among participants. Similar to the competitors, we have detected a spike on 2004-10-23, corresponding to the annual sponsor meeting that occurred on 2004-10-21. More than two-thirds of the participants have attended the meeting, focusing on achieving project goals throughout the week [EP06]. Moreover, we have detected other change points related to national holidays and spring break.

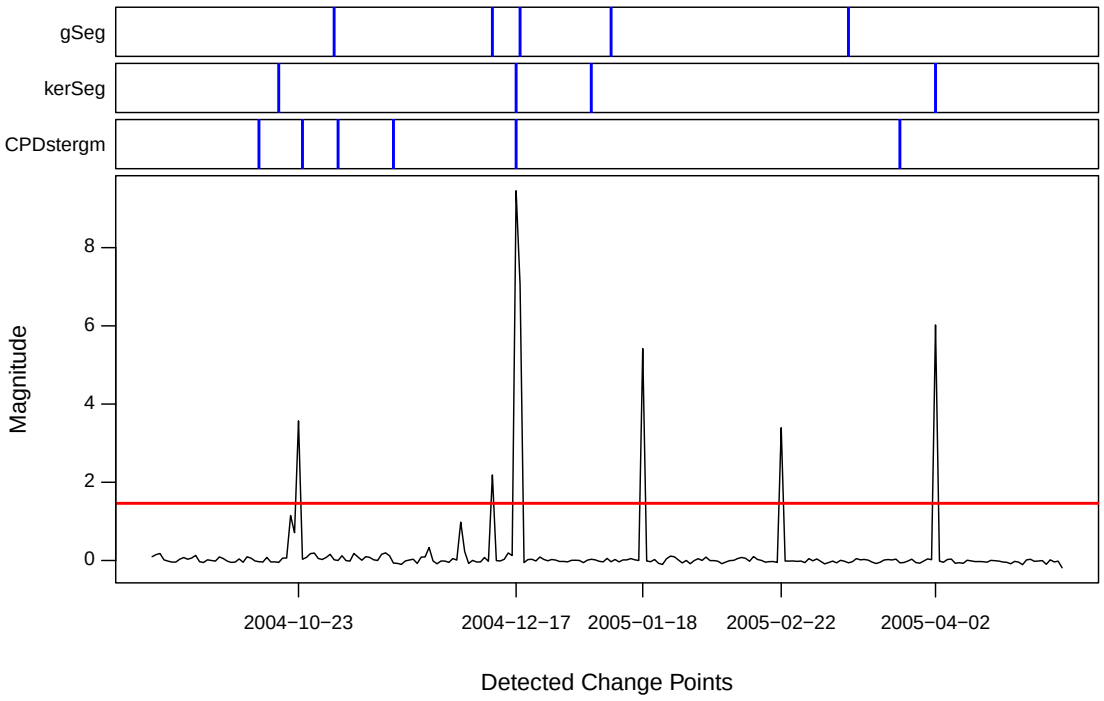

Figure 3.5: Detected change points from the proposed and competitor (blue) methods on the MIT Cellphone Data. The threshold (red horizontal line) is calculated by (3.11) with $\mathcal{Z}_{0.9}$.

3.5.3 Enron Email Data

The Enron email data, analyzed by [PCM05, PPY12, PC15], portrays communication among employees, before the collapse of a giant energy company. We construct $T = 100$ weekly networks, ranging from 2000-06-05 to 2002-05-06 for $n = 100$ employees. We detect the change points with our method, and we use the same network statistics described in Section 3.5.2 to the competitor methods. Figure 3.6 displays $\Delta\hat{\boldsymbol{\zeta}}$ of Equation (3.12), and the detected change points from our method and competitor methods. Furthermore, Table 3.5 provides a list of potential events, aligning with the detected change points from our method.

In 2001, Enron underwent a multitude of major incidents, making it difficult to associate

Table 3.4: Potential nearby events aligned with the detected change points (CP) from our proposed method on the MIT cellphone data.

Detected CP	Potential nearby events
2004-10-23	2004-10-21 Sponsor meeting
2004-12-17	2004-12-18 to 2005-01-02 Winter break
2005-01-18	2005-01-17 Martin Luther King Day
2005-02-22	2005-02-21 Presidents Day
2005-04-02	2005-03-21 to 2005-03-25 Spring break

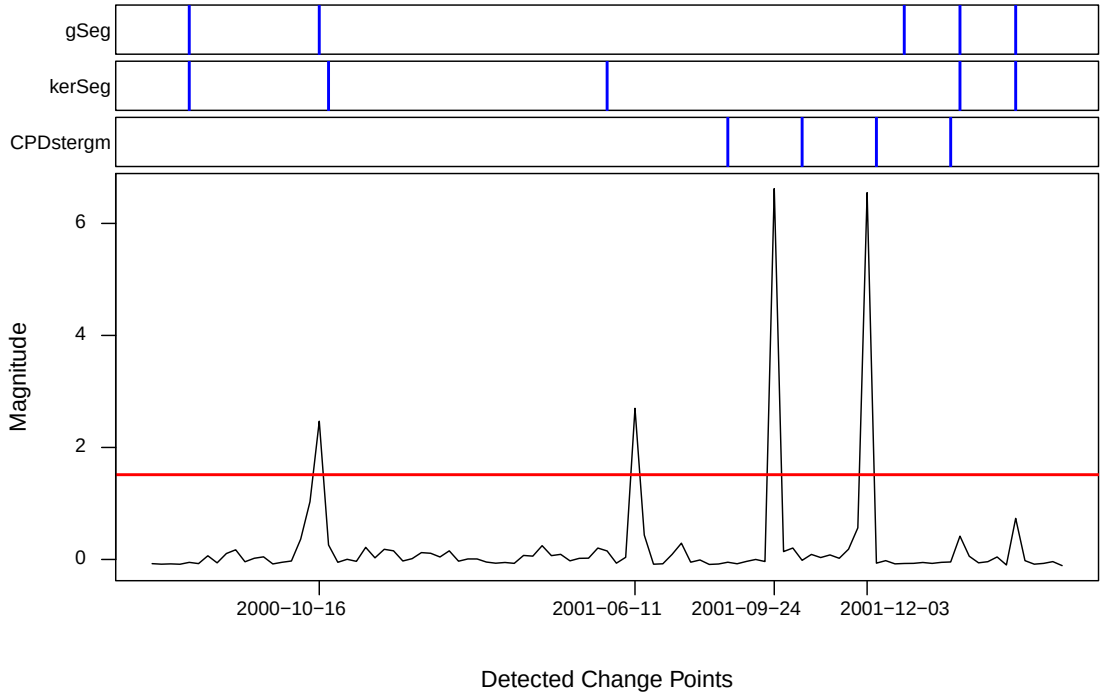

Figure 3.6: Detected change points from the proposed and competitor (blue) methods on the Enron email data. The threshold (red horizontal line) is calculated by (3.11) with $\mathcal{Z}_{0.9}$.

Table 3.5: Potential nearby events aligned with the detected change points (CP) from our proposed method on the Enron email data.

Detected CP	Potential nearby events
2000-10-16	2000-11-01 FERC exonerated Enron
2001-06-11	2001-06-21 CEO publicly confronted
2001-09-24	2001-08-14 CEO resigned
2001-12-03	2001-12-02 Enron filed for bankruptcy

the detected change points with real world events, locally. Yet, as our proposed method provides global results over the entire time frame, four crucial change points are detected for interpretation. Throughout 2000, Enron orchestrated rolling blackouts, causing staggering surges in electricity prices that peaked at twenty times the standard rate. The situation worsened when the Federal Energy Regulatory Commission (FERC) exonerated Enron of wrongdoing by the end of 2000. Subsequently, an activist physically confronted the CEO in protest against Enron's role in the energy crisis, and Enron's stock price plummeted after the CEO's resignation in August 2001. Three months later, pressured by Wall Street analysts and the revelation of the scandals, Enron filed for bankruptcy and the largest energy company in the U.S. fell apart.

3.6 Discussion

This chapter proposes a simple generative model to detect change points in dynamic graphs. Intrinsically, dynamic networks are complex due to dyadic and temporal dependencies. Learning low dimensional graph representations can extract useful features serving change point detection. We impose prior distributions to the graph representations, and the informative priors in the latent space are learned from data as empirical Bayes. The Group Fused

Lasso problem is solved via ADMM, and the generative model is demonstrated to be helpful for change point detection.

Several extensions to our proposed framework are possible for future development. Besides binary networks, relations by nature have degree of strength, which are denoted by generic values. Moreover, nodal and dyadic attributes are important components in network data. Hence, models that generate weighted edges, as well as nodal and dyadic attributes, can capture more information about the networks [FH12b, Kri12b]. While our framework demonstrates the ability in change point detection, the development of more sophisticated architectures can enhance the model's capacity on other tasks and data [HRT07, KSA10, YMW21, MXW23].

3.7 Appendix

3.7.1 Technical Details

3.7.1.1 Updating $\boldsymbol{\mu}$ and $\boldsymbol{\phi}$

In this section, we derive the updates for prior parameter $\boldsymbol{\mu} \in \mathbb{R}^{T\times d}$ and graph decoder parameter $\boldsymbol{\phi}$. Denote the objective function in Equation (3.4) as $\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\mu})$ and denote the set of parameters $\{\boldsymbol{\phi}, \boldsymbol{\mu}\}$ as $\boldsymbol{\theta}$. We first calculate the gradient of the log-likelihood $l(\boldsymbol{\theta})$ in

$\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\mu})$ with respect to $\boldsymbol{\theta}$:

$$
\begin{aligned}
\nabla_{\boldsymbol{\theta}}\ l(\boldsymbol{\theta}) &= \nabla_{\boldsymbol{\theta}} \sum_{t=1}^{T} \log P(\boldsymbol{y}^t) \\
&= \sum_{t=1}^{T} \frac{1}{P(\boldsymbol{y}^t)} \nabla_{\boldsymbol{\theta}} P(\boldsymbol{y}^t) \\
&= \sum_{t=1}^{T} \frac{1}{P(\boldsymbol{y}^t)} \nabla_{\boldsymbol{\theta}} \int P(\boldsymbol{y}^t, \boldsymbol{z}^t) d\boldsymbol{z}^t \\
&= \sum_{t=1}^{T} \frac{1}{P(\boldsymbol{y}^t)} \int P(\boldsymbol{y}^t, \boldsymbol{z}^t) \Big[\nabla_{\boldsymbol{\theta}} \log P(\boldsymbol{y}^t, \boldsymbol{z}^t) \Big] d\boldsymbol{z}^t \\
&= \sum_{t=1}^{T} \int \frac{P(\boldsymbol{y}^t, \boldsymbol{z}^t)}{P(\boldsymbol{y}^t)} \Big[\nabla_{\boldsymbol{\theta}} \log P(\boldsymbol{y}^t, \boldsymbol{z}^t) \Big] d\boldsymbol{z}^t \\
&= \sum_{t=1}^{T} \int P(\boldsymbol{z}^t | \boldsymbol{y}^t) \Big[\nabla_{\boldsymbol{\theta}} \log P(\boldsymbol{y}^t, \boldsymbol{z}^t) \Big] d\boldsymbol{z}^t \\
&= \sum_{t=1}^{T} \mathbb{E}_{P(\boldsymbol{z}^t | \boldsymbol{y}^t)} \Big(\nabla_{\boldsymbol{\theta}} \log \Big[P(\boldsymbol{y}^t | \boldsymbol{z}^t) P(\boldsymbol{z}^t) \Big] \Big) \\
&= \sum_{t=1}^{T} \mathbb{E}_{P(\boldsymbol{z}^t | \boldsymbol{y}^t)} \Big(\nabla_{\boldsymbol{\theta}} \log P(\boldsymbol{y}^t | \boldsymbol{z}^t) \Big) + \sum_{t=1}^{T} \mathbb{E}_{P(\boldsymbol{z}^t | \boldsymbol{y}^t)} \Big(\nabla_{\boldsymbol{\theta}} \log P(\boldsymbol{z}^t) \Big).
\end{aligned}
$$

Note that the expectation in the gradient is now with respect to the posterior distribution $P(\boldsymbol{z}^t | \boldsymbol{y}^t) \propto P(\boldsymbol{y}^t | \boldsymbol{z}^t) \times P(\boldsymbol{z}^t)$. Furthermore, the gradient of $\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\mu})$ with respect to the prior parameter $\boldsymbol{\mu}^t \in \mathbb{R}^d$ at a specific time point t is

$$
\begin{aligned}
\nabla_{\boldsymbol{\mu}^t}\ \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\mu}) &= -\mathbb{E}_{P(\boldsymbol{z}^t | \boldsymbol{y}^t)} \Big(\nabla_{\boldsymbol{\mu}^t} \log P(\boldsymbol{z}^t) \Big) + \kappa(\boldsymbol{\mu}^t - \boldsymbol{\nu}^t + \boldsymbol{w}^t) \\
&= -\mathbb{E}_{P(\boldsymbol{z}^t | \boldsymbol{y}^t)} (\boldsymbol{z}^t - \boldsymbol{\mu}^t) + \kappa(\boldsymbol{\mu}^t - \boldsymbol{\nu}^t + \boldsymbol{w}^t).
\end{aligned}
$$

Setting the gradient $\nabla_{\boldsymbol{\mu}^t}\ \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\mu})$ to zeros and solve for $\boldsymbol{\mu}^t$, we have

$$
\begin{aligned}
\mathbf{0} &= -\mathbb{E}_{P(\boldsymbol{z}^t | \boldsymbol{y}^t)} (\boldsymbol{z}^t) + \boldsymbol{\mu}^t + \kappa \boldsymbol{\mu}^t - \kappa \boldsymbol{\nu}^t + \kappa \boldsymbol{w}^t \\
\mathbf{0} &= -\mathbb{E}_{P(\boldsymbol{z}^t | \boldsymbol{y}^t)} (\boldsymbol{z}^t) + (1 + \kappa) \boldsymbol{\mu}^t - \kappa(\boldsymbol{\nu}^t - \boldsymbol{w}^t) \\
(1 + \kappa) \boldsymbol{\mu}^t &= \mathbb{E}_{P(\boldsymbol{z}^t | \boldsymbol{y}^t)} (\boldsymbol{z}^t) + \kappa(\boldsymbol{\nu}^t - \boldsymbol{w}^t) \\
\boldsymbol{\mu}^t &= \frac{1}{1 + \kappa} \mathbb{E}_{P(\boldsymbol{z}^t | \boldsymbol{y}^t)} (\boldsymbol{z}^t) + \frac{\kappa}{1 + \kappa} (\boldsymbol{\nu}^t - \boldsymbol{w}^t)
\end{aligned}
$$

concluding Proposition 1. Evidently, the gradient of $\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\mu})$ with respect to the graph decoder parameter $\boldsymbol{\phi}$ is

$$\nabla_{\boldsymbol{\phi}}\ \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\mu}) = -\sum_{t=1}^{T} \mathbb{E}_{P(\boldsymbol{z}^t|\boldsymbol{y}^t)}\Big(\nabla_{\boldsymbol{\phi}} \log P(\boldsymbol{y}^t|\boldsymbol{z}^t)\Big).$$

3.7.1.2 Langevin Dynamics

Calculating the solution in (3.7) and the gradient in (3.8) requires evaluating the conditional expectations under the posterior distribution $P(\boldsymbol{z}^t|\boldsymbol{y}^t) \propto P(\boldsymbol{y}^t|\boldsymbol{z}^t) \times P(\boldsymbol{z}^t)$. In this section, we discuss the Langevin Dynamics to sample $\boldsymbol{z}^t \in \mathbb{R}^d$ from the posterior distribution $P(\boldsymbol{z}^t|\boldsymbol{y}^t)$ that is conditional on the observed network $\boldsymbol{y}^t \in \{0,1\}^{n\times n}$. In particular, the Langevin Dynamics, a short run MCMC, is achieved by iterating the following:

$$\begin{aligned}
\boldsymbol{z}_{l+1}^t &= \boldsymbol{z}_l^t + s\nabla_{\boldsymbol{z}^t} \log P(\boldsymbol{z}^t|\boldsymbol{y}^t) + \sqrt{2s}\boldsymbol{\epsilon} \\
&= \boldsymbol{z}_l^t + s\nabla_{\boldsymbol{z}^t} \log\Big[\frac{P(\boldsymbol{y}^t|\boldsymbol{z}^t)P(\boldsymbol{z}^t)}{P(\boldsymbol{y}^t)}\Big] + \sqrt{2s}\boldsymbol{\epsilon} \\
&= \boldsymbol{z}_l^t + s\big[\nabla_{\boldsymbol{z}^t} \log P(\boldsymbol{y}^t|\boldsymbol{z}^t) + \nabla_{\boldsymbol{z}^t} \log P(\boldsymbol{z}^t) - \nabla_{\boldsymbol{z}^t} \log P(\boldsymbol{y}^t)\big] + \sqrt{2s}\boldsymbol{\epsilon} \\
&= \boldsymbol{z}_l^t + s\big[\nabla_{\boldsymbol{z}^t} \log P(\boldsymbol{y}^t|\boldsymbol{z}^t) - (\boldsymbol{z}_l^t - \boldsymbol{\mu}^t)\big] + \sqrt{2s}\boldsymbol{\epsilon}
\end{aligned}$$

where l is the time step and s is the step size of the Langevin Dynamics. The error term $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$ serves as a random perturbation to the sampling process. In Section 3.5, we set $l = 40$, $s = 0.5$, and we draw $m = 100$ samples for each time point $t = 1, \ldots, T$ within an ADMM iteration. The gradient of the graph decoder $P(\boldsymbol{y}^t|\boldsymbol{z}^t)$ with respect to the latent variable $\boldsymbol{z}^t$ can be calculated efficiently through back-propagation. Consequently, we use the MCMC samples to approximate the conditional expectation $\mathbb{E}_{P(\boldsymbol{z}^t|\boldsymbol{y}^t)}(\cdot)$ in the solution (3.7) and the gradient (3.8).

3.7.1.3 Group Lasso for Updating $\boldsymbol{\beta}$

In this section, we present the derivation to update $\boldsymbol{\beta}$ in Proposition 2, which is equivalent to solving a Group Lasso problem [YL06]. We adapt the derivation from [BV11] for our

proposed ADMM algorithm. Denote the objective function in (3.5) as $\mathcal{L}(\boldsymbol{\gamma}, \boldsymbol{\beta})$. When $\boldsymbol{\beta}_{t,\cdot} \neq \mathbf{0}$, the gradient of $\mathcal{L}(\boldsymbol{\gamma}, \boldsymbol{\beta})$ with respect to $\boldsymbol{\beta}_{t,\cdot}$ is

$$\nabla_{\boldsymbol{\beta}_{t,\cdot}} \mathcal{L}(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \lambda \frac{\boldsymbol{\beta}_{t,\cdot}}{\|\boldsymbol{\beta}_{t,\cdot}\|_2} - \kappa \boldsymbol{X}_{\cdot,t}^\top (\boldsymbol{\mu}_{(a+1)} + \boldsymbol{w}_{(a)} - \mathbf{1}_{T,1}\boldsymbol{\gamma} - \boldsymbol{X}_{\cdot,t}\boldsymbol{\beta}_{t,\cdot} - \boldsymbol{X}_{\cdot,-t}\boldsymbol{\beta}_{-t,\cdot})$$

where $\boldsymbol{X}_{\cdot,t} \in \mathbb{R}^{T\times 1}$ is the t-th column of matrix $\boldsymbol{X} \in \mathbb{R}^{T\times(T-1)}$ and $\boldsymbol{\beta}_{t,\cdot} \in \mathbb{R}^{1\times d}$ is the t-th row of matrix $\boldsymbol{\beta} \in \mathbb{R}^{(T-1)\times d}$. Moreover, we denote $\boldsymbol{\beta}_{-t,\cdot} \in \mathbb{R}^{(T-1)\times p}$ as the matrix obtained by replacing the t-th row of matrix $\boldsymbol{\beta}$ with a zero vector, and $\boldsymbol{X}_{\cdot,-t} \in \mathbb{R}^{T\times(T-1)}$ is denoted similarly.

Setting the above gradient to zeros, we have

$$\boldsymbol{\beta}_{t,\cdot} = (\kappa \boldsymbol{X}_{\cdot,t}^\top \boldsymbol{X}_{\cdot,t} + \frac{\lambda}{\|\boldsymbol{\beta}_{t,\cdot}\|_2})^{-1} \boldsymbol{b}_t \tag{3.13}$$

where

$$\boldsymbol{b}_t = \kappa \boldsymbol{X}_{\cdot,t}^\top (\boldsymbol{\mu}_{(a+1)} + \boldsymbol{w}_{(a)} - \mathbf{1}_{T,1}\boldsymbol{\gamma} - \boldsymbol{X}_{\cdot,-t}\boldsymbol{\beta}_{-t,\cdot}) \in \mathbb{R}^{1\times d}.$$

Calculating the Euclidean norm of (3.13) on both sides and rearrange the terms, we have

$$\|\boldsymbol{\beta}_{t,\cdot}\|_2 = (\kappa \boldsymbol{X}_{\cdot,t}^\top \boldsymbol{X}_{\cdot,t})^{-1}(\|\boldsymbol{b}_t\|_2 - \lambda).$$

Plugging $\|\boldsymbol{\beta}_{t,\cdot}\|_2$ into (3.13) for substitution, the solution of $\boldsymbol{\beta}_{t,\cdot}$ is arrived at

$$\boldsymbol{\beta}_{t,\cdot} = \frac{1}{\kappa \boldsymbol{X}_{\cdot,t}^\top \boldsymbol{X}_{\cdot,t}}(1 - \frac{\lambda}{\|\boldsymbol{b}_t\|_2})\boldsymbol{b}_t.$$

Moreover, when $\boldsymbol{\beta}_{t,\cdot} = \mathbf{0}$, the subgradient $\boldsymbol{v}$ of $\|\boldsymbol{\beta}_{t,\cdot}\|_2$ needs to satisfy that $\|\boldsymbol{v}\|_2 \leq 1$. Because

$$\mathbf{0} \in \lambda \boldsymbol{v} - \kappa \boldsymbol{X}_{\cdot,t}^\top (\boldsymbol{\mu}_{(a+1)} + \boldsymbol{w}_{(a)} - \mathbf{1}_{T,1}\boldsymbol{\gamma} - \boldsymbol{X}_{\cdot,-t}\boldsymbol{\beta}_{-t,\cdot}),$$

we obtain the condition that $\boldsymbol{\beta}_{t,\cdot}$ becomes $\mathbf{0}$ when $\|\boldsymbol{b}_t\|_2 \leq \lambda$. Therefore, we can iteratively apply the following to update $\boldsymbol{\beta}_{t,\cdot}$ for each block $t = 1, \ldots, T-1$:

$$\boldsymbol{\beta}_{t,\cdot} \leftarrow \frac{1}{\kappa \boldsymbol{X}_{\cdot,t}^\top \boldsymbol{X}_{\cdot,t}} \Big(1 - \frac{\lambda}{\|\boldsymbol{b}_t\|_2}\Big)_+ \boldsymbol{b}_t$$

where $(\cdot)_+ = \max(\cdot, 0)$.

3.7.2 Practical Guidelines

3.7.2.1 ADMM Implementation

In this section, we provide practical guidelines for the proposed framework and the Alternating Direction Method of Multipliers (ADMM) algorithm. To detect the change points with our method in the simulation study, we let the latent dimensions $d = 10$ and $k = 5$ for the graph decoder that is defined in Section 3.2.2. Furthermore, we initialize the penalty parameter $\kappa = 10$, and we let the tuning parameter $\lambda = \{10, 20, 50, 100\}$. For each λ, we run $A = 50$ iterations of ADMM. Within each ADMM iteration, we run $B = 20$ iterations of gradient descent with Adam optimizer for the graph decoder and $D = 20$ iterations of block coordinate descent for Group Lasso. To construct the data-driven threshold ϵ_{thr} in (3.11), we use the 90% quantile of the standard Normal distribution. The 3-layer MLP in the graph decoder is trained on a Tesla T4 GPU, and the computing time is about 1 hour for 10 Monte Carlo trials in each simulation study. The code to reproduce the result in the manuscript is provided in the Appendix.

Since the proposed generative model is a probability distribution for the observed network data, in this work we stop ADMM learning with the following stopping criteria:

$$\left| \frac{l(\boldsymbol{\phi}_{(a+1)}, \boldsymbol{\mu}_{(a+1)}) - l(\boldsymbol{\phi}_{(a)}, \boldsymbol{\mu}_{(a)})}{l(\boldsymbol{\phi}_{(a)}, \boldsymbol{\mu}_{(a)})} \right| \leq \epsilon_{\text{tol}}. \tag{3.14}$$

The log-likelihood $l(\boldsymbol{\phi}, \boldsymbol{\mu})$ is approximated by sampling from the prior distribution $p(\boldsymbol{z}^t)$, as described in Section 3.4.2. Hence, we stop the ADMM procedure until the above criteria is satisfied for a' consecutive iterations. In Section 3.5, we set $\epsilon_{\text{tol}} = 10^{-5}$ and $a' = 5$.

Here we briefly elaborate on the computational aspect of the approximation of the log-likelihood. To calculate the product of edge probabilities for the conditional distribution

$P(\boldsymbol{y}^t|\boldsymbol{z}^t)$, we have the following:

$$\begin{aligned}
\sum_{t=1}^{T} \log P(\boldsymbol{y}^t) &= \sum_{t=1}^{T} \log \int P(\boldsymbol{y}^t|\boldsymbol{z}^t) P(\boldsymbol{z}^t) d\boldsymbol{z}^t \\
&= \sum_{t=1}^{T} \log \mathbb{E}_{P(\boldsymbol{z}^t)}[\prod_{(i,j)\in\mathbb{Y}} P(\boldsymbol{y}_{ij}^t|\boldsymbol{z}^t)] \\
&\approx \sum_{t=1}^{T} \log \Big[\frac{1}{m}\sum_{u=1}^{m}[\prod_{(i,j)\in\mathbb{Y}} P(\boldsymbol{y}_{ij}^t|\boldsymbol{z}_u^t)]\Big] \\
&= \sum_{t=1}^{T} \log \Big[\frac{1}{m}\sum_{u=1}^{m} \exp\{\sum_{(i,j)\in\mathbb{Y}} \log[P(\boldsymbol{y}_{ij}^t|\boldsymbol{z}_u^t)]\}\Big] \\
&= \sum_{t=1}^{T} \Big\{ -\log m + \log\Big[\exp C^t \sum_{u=1}^{m} \exp\{\sum_{(i,j)\in\mathbb{Y}} \log[P(\boldsymbol{y}_{ij}^t|\boldsymbol{z}_u^t)] - C^t\}\Big]\Big\} \\
&= \sum_{t=1}^{T} \Big\{ C^t + \log\Big[\sum_{u=1}^{m} \exp\{\sum_{(i,j)\in\mathbb{Y}} \log[P(\boldsymbol{y}_{ij}^t|\boldsymbol{z}_u^t)] - C^t\}\Big]\Big\} - T\log m
\end{aligned}$$

where $C^t \in \mathbb{R}$ is the maximum value of $\sum_{(i,j)\in\mathbb{Y}} \log[P(\boldsymbol{y}_{ij}^t|\boldsymbol{z}_u^t)]$ over m samples but within a time point t.

We also update the penalty parameter κ to improve convergence and to reduce reliance on its initialization. In particular, after the a-th ADMM iteration, we calculate the respective primal and dual residuals:

$$\begin{aligned}
r_{\text{primal}}^{(a)} &= \sqrt{\frac{1}{T\times d}\sum_{t=1}^{T} \|\boldsymbol{\mu}_{(a)}^t - \boldsymbol{\nu}_{(a)}^t\|_2^2}, \\
r_{\text{dual}}^{(a)} &= \sqrt{\frac{1}{T\times d}\sum_{t=1}^{T} \|\boldsymbol{\nu}_{(a)}^t - \boldsymbol{\nu}_{(a-1)}^t\|_2^2}.
\end{aligned}$$

Throughout, we jointly update the penalty parameter $\kappa \in \mathbb{R}$ and the scaled dual variable $\boldsymbol{w} \in \mathbb{R}^{T\times d}$ as in [BPC11] with the following conditions:

$$\begin{aligned}
\kappa_{(a+1)} &= 2\kappa_{(a)}, \quad \boldsymbol{w}_{(a+1)} = \frac{1}{2}\boldsymbol{w}_{(a)} \quad \text{if } r_{\text{primal}}^{(a)} > 10 \times r_{\text{dual}}^{(a)}, \\
\kappa_{(a+1)} &= \frac{1}{2}\kappa_{(a)}, \quad \boldsymbol{w}_{(a+1)} = 2\boldsymbol{w}_{(a)} \quad \text{if } r_{\text{dual}}^{(a)} > 10 \times r_{\text{primal}}^{(a)}.
\end{aligned}$$

3.7.2.2 Post-Processing

Since neural networks may be over-fitted for a statistical model in change point detection, we track the following Coefficient of Variation as a signal-to-noise ratio when we learn the model parameter with the full data:

$$\text{Coefficient of Variation} = \frac{\text{mean}(\Delta\hat{\boldsymbol{\mu}})}{\text{sd}(\Delta\hat{\boldsymbol{\mu}})}.$$

We choose the learned parameter $\hat{\boldsymbol{\mu}}$ with the largest Coefficient of Variation as final output.

By convention, we also implement two post-processing steps to finalize the detected change points. When the gap between two consecutive change points is small or $\hat{C}_k - \hat{C}_{k-1} < \epsilon_{\text{spc}}$, we preserve the detected change point with greater $\Delta\hat{\boldsymbol{\zeta}}$ value to prevent clusters of nearby change points. Moreover, as the endpoints of a time span are usually not of interest, we remove the $\hat{C}_k$ smaller than a threshold ϵ_{end} and the $\hat{C}_k$ greater than $T - \epsilon_{\text{end}}$. In Section 3.5, we set $\epsilon_{\text{spc}} = 5$ and $\epsilon_{\text{end}} = 5$.

CHAPTER 4

A Partially Separable Model for Dynamic Valued Networks

The Exponential-family Random Graph Model (ERGM) is a powerful model to fit networks with complex structures. However, for dynamic valued networks whose observations are matrices of counts that evolve over time, the development of the ERGM framework is still in its infancy. To facilitate the modeling of dyad value increment and decrement, a Partially Separable Temporal ERGM is proposed for dynamic valued networks. The parameter learning algorithms inherit state-of-the-art estimation techniques to approximate the maximum likelihood, by drawing Markov chain Monte Carlo (MCMC) samples conditioning on the valued network from the previous time step. The ability of the proposed model to interpret network dynamics and forecast temporal trends is demonstrated with real data.

4.1 Introduction

Networks are used to represent relational phenomena in many domains, such as stock relations in financial market [FHW19], scene graphs in computer vision [SMS21], and mitochondrial networks in cancer metabolism [HBS23]. Conventionally, relations are indicated by the presence or absence of ties. Though connected ties are seemingly identical, relations by nature have degree of strength, which can be represented by generic values to distinguish them. Often, valued networks are dichotomized into binary networks for analysis, which curtails the information that original networks convey. To prevent potential bias from

data thresholding [TB11], [Kri12a] extended the Exponential-family Random Graph Model (ERGM) to fit networks with count dyad values. [DC12a] and [WDB17] focused on networks with continuous-valued edges. Moreover, [CG20] proposed to model weighted networks in a hierarchical multilayer framework.

Relational phenomena also progress in time. [RP01] first proposed to model dynamic networks in a Markovian discrete time framework. [Sni01] and [Sni05] developed a Stochastic Actor-Oriented Model, which is driven by the actor's perspective to make or withdraw ties to other actors. [But08b] introduced a Relational Event Model, focusing on the action emitted by an entity toward another. Furthermore, [HFX10] defined a Temporal ERGM (TERGM), by specifying a conditional ERGM between consecutive networks. We refer the interested reader to [SM17] for a review in modeling network dynamics.

In this chapter, we focus on the structure of count valued networks over time. Besides being limited to binary networks, existing frameworks model snapshots of networks, which gives little insight into the underlying dynamic process and little prediction power in how future networks will evolve [JLY20, GD20]. While a snapshot of a valued network presents the structural properties appearing at the observed time point, it does not provide information about the dynamics that produce the structural properties, such as the amount and rate at which the dyad values increase or decrease. Moreover, as we will demonstrate below, without a decomposition that separates dyad value increment and decrement, interpreting network dynamics can be challenging.

[YCZ11] proposed a dynamic stochastic block model, by capturing the transition of community memberships for individual nodes. [SC16] proposed a latent space model, by assuming the probability of a stronger edge between two nodes is greater when they are closer in the latent space. These models provide a good comprehension of the relations between actors over time, though they may not extend to other network structures of interest that signify the generating process. Furthermore, [WCB10] included a dynamic feature term in ERGM to capture the change in dyad values over time. Yet the interpretation of network

dynamics may be difficult.

The ERGM that defines local forces to shape global structure [HHB08b] is a natural way to model complex networks. Inspired by [KH14] in modeling dynamic binary networks with a Separable Temporal ERGM (STERGM), we extend the ERGM framework to model dynamic valued networks as follows.

- We propose a Partially Separable Temporal ERGM (PST ERGM) to fit dynamic valued networks, assuming the factors that increase relational strength are different from those that decrease relational strength. In particular, we construct two intermediate networks to manage dyad value increment and decrement, separately. The dynamics are specified with two sets of network statistics evaluated on the intermediate networks, and we use two sets of parameters to facilitate interpretation.

- We adapt recent advances in fitting static binary networks from [HHH12b] to seed an initial configuration for parameter learning. We exploit the Contrastive Divergence sampling, an abridged MCMC from [Hum11] and [Kri17b], to expedite the learning process. We also provide the Metropolis-Hastings algorithm to sample dynamic valued networks conditioning on the previous networks.

- Our experiments show a good performance of the proposed model. In particular, a time-heterogeneous PST ERGM on the students contact networks [MFB15] provides a realistic interpretation of the network dynamics. Furthermore, a time-homogeneous PST ERGM on the baboons interaction networks [GGP20] produces reasonable out-of-sample forecasts of the temporal trends.

The rest of the chapter is organized as follows. In Section 4.2, we review ERGM for static binary and valued networks, and STERGM for dynamic binary networks. In Section 4.3, we propose the PST ERGM for dynamic valued networks with specifications on the intermediate networks. In Section 4.4, we discuss the approximate maximum likelihood estimation and

the Metropolis-Hastings algorithm for drawing dynamic valued networks. In Section 4.5, we illustrate the methodology with simulated and real data examples. In Section 4.6, we conclude with a discussion and potential future developments.

4.2 ERGM for Networks

4.2.1 ERGM for Static Binary and Valued Networks

For a fixed set $N = \{1, 2, \cdots, n\}$ of nodes, we can use a network $\boldsymbol{y} \in \mathcal{Y}$, in the form of an $n \times n$ matrix, to represent the potential relations for all pairs $(i, j) \in \mathbb{Y} \subseteq N \times N$. The binary networks have dyad $\boldsymbol{y}_{ij} \in \{0, 1\}$ to represent the absence or presence of a tie and $\mathcal{Y} \subseteq 2^{\mathbb{Y}}$. Let $\mathbb{N}_0$ be the set of natural numbers and 0. The valued networks have dyad $\boldsymbol{y}_{ij} \in \mathbb{N}_0$ to represent the intensity of a tie and $\mathcal{Y} \subseteq \mathbb{N}_0^{\mathbb{Y}}$. We disallow a network to have self-edge, so $\boldsymbol{y}_{ij} = 0$ if $i = j$. The relation in a network can be either directed or undirected, where an undirected network has $\boldsymbol{y}_{ij} = \boldsymbol{y}_{ji}$ for all dyads (i, j). In this chapter, we focus on undirected networks, and the directed variant follows naturally.

The probabilistic formulation of an ERGM for a network $\boldsymbol{y}$ is

$$P(\boldsymbol{y}; \boldsymbol{\eta}) = h(\boldsymbol{y}) \exp[\boldsymbol{\eta}^\top \boldsymbol{g}(\boldsymbol{y}) - \psi(\boldsymbol{\eta})],$$

where $\boldsymbol{g}(\boldsymbol{y})$, with $\boldsymbol{g} : \mathcal{Y} \to \mathbb{R}^p$, is a vector of network statistics; $\boldsymbol{\eta} \in \mathbb{R}^p$ is a vector of unknown parameters; $\exp[\psi(\boldsymbol{\eta})] = \sum_{\boldsymbol{y} \in \mathcal{Y}} h(\boldsymbol{y}) \exp[\boldsymbol{\eta}^\top \boldsymbol{g}(\boldsymbol{y})]$ is the normalizing constant; $h(\boldsymbol{y})$, with $h : \mathcal{Y} \to [0, \infty)$, is the reference function. Moreover, the network statistics $\boldsymbol{g}(\boldsymbol{y})$ may also depend on nodal attributes $\boldsymbol{x}$ and dyadic attributes $\boldsymbol{z}$. For notational simplicity, we omit the dependence of $\boldsymbol{g}(\boldsymbol{y})$ on $\boldsymbol{x}$ and $\boldsymbol{z}$.

In valued ERGM [Kri12a], the parameter space of $\boldsymbol{\eta}$ has to ensure that $P(\boldsymbol{y}; \boldsymbol{\eta})$ is a valid probability distribution. When the range of dyad value in a network is $\mathbb{N}_0$, the condition $\exp[\psi(\boldsymbol{\eta})] < \infty$ is sufficient to guarantee that $P(\boldsymbol{y}; \boldsymbol{\eta})$ is a valid distribution. Furthermore, the reference function $h(\boldsymbol{y})$ underlies a baseline distribution of dyad values. That is when

$\boldsymbol{\eta} = \mathbf{0}$, $P(\boldsymbol{y}; \boldsymbol{\eta}) \propto h(\boldsymbol{y})$. Specifically, [Kri12a] defined a Poisson-reference ERGM and [Kri19] defined a Binomial-reference ERGM for valued networks with respective reference functions:

$$h_{\text{Pois}}(\boldsymbol{y}) = \prod_{(i,j)\in\mathbb{Y}} (\boldsymbol{y}_{ij}!)^{-1} \quad \text{and} \quad h_{\text{Bino}}(\boldsymbol{y}) = \prod_{(i,j)\in\mathbb{Y}} \binom{m}{\boldsymbol{y}_{ij}},$$

where m is a known maximum value that each relationship $\boldsymbol{y}_{ij} \in \{0, 1, \cdots, m\}$ can take in this network $\boldsymbol{y}$. For binary ERGM, usually $h(\boldsymbol{y}) = 1$ [WP96, Sni02, SPR06, HH06b].

4.2.2 STERGM for Dynamic Binary Networks

The TERGM [HFX10] for a binary network $\boldsymbol{y}^t$ conditional on $\boldsymbol{y}^{t-1}$ is

$$P(\boldsymbol{y}^t | \boldsymbol{y}^{t-1}; \boldsymbol{\eta}) = \exp[\boldsymbol{\eta}^\top \boldsymbol{g}(\boldsymbol{y}^t, \boldsymbol{y}^{t-1}) - \psi(\boldsymbol{\eta}, \boldsymbol{y}^{t-1})],$$

where $\boldsymbol{y}^t \in \mathcal{Y}^t \subseteq 2^{\mathbb{Y}}$ is a single network at a discrete time point t. The $\boldsymbol{g}(\boldsymbol{y}^t, \boldsymbol{y}^{t-1})$, with $\boldsymbol{g} : \mathcal{Y}^t \times \mathcal{Y}^{t-1} \to \mathbb{R}^p$, is a vector of network statistics for the transition from $\boldsymbol{y}^{t-1}$ to $\boldsymbol{y}^t$. Yet [KH14] demonstrated that higher coefficients in TERGM can lead to inconsistent interpretation of network evolution in terms of incidence and duration. Hence a careful decomposition of network dynamics is needed. In particular, the incidence, how often new ties form, can be measured by dyad formation, and the duration, how long old ties last, can be measured by dyad dissolution.

Instead of modeling the observed $\boldsymbol{y}^t$ given $\boldsymbol{y}^{t-1}$ that muddles network dynamics, [KH14] designed two intermediate networks, the formation network and dissolution network, between time $t-1$ and t to reflect the incidence and duration. The formation network $\boldsymbol{y}^{+,t} \in \mathcal{Y}^{+,t}$ is acquired by adding the edges formed at time t to $\boldsymbol{y}^{t-1}$ so $\mathcal{Y}^{+,t} \subseteq \{\boldsymbol{y} \in 2^{\mathbb{Y}} : \boldsymbol{y} \supseteq \boldsymbol{y}^{t-1}\}$. The dissolution network $\boldsymbol{y}^{-,t} \in \mathcal{Y}^{-,t}$ is acquired by removing the edges dissolved at time t from $\boldsymbol{y}^{t-1}$ so $\mathcal{Y}^{-,t} \subseteq \{\boldsymbol{y} \in 2^{\mathbb{Y}} : \boldsymbol{y} \subseteq \boldsymbol{y}^{t-1}\}$.

Assuming $\boldsymbol{y}^{+,t}$ is conditionally independent of $\boldsymbol{y}^{-,t}$ given $\boldsymbol{y}^{t-1}$, the STERGM [KH14] for $\boldsymbol{y}^t$ conditional on $\boldsymbol{y}^{t-1}$ is

$$P(\boldsymbol{y}^t | \boldsymbol{y}^{t-1}; \boldsymbol{\eta}^+, \boldsymbol{\eta}^-) = P(\boldsymbol{y}^{+,t} | \boldsymbol{y}^{t-1}; \boldsymbol{\eta}^+) \times P(\boldsymbol{y}^{-,t} | \boldsymbol{y}^{t-1}; \boldsymbol{\eta}^-), \tag{4.1}$$

with the respective formation model and dissolution model specified as

$$P(\boldsymbol{y}^{+,t}|\boldsymbol{y}^{t-1};\boldsymbol{\eta}^{+}) = \exp[\boldsymbol{\eta}^{+}\cdot\boldsymbol{g}^{+}(\boldsymbol{y}^{+,t},\boldsymbol{y}^{t-1}) - \psi(\boldsymbol{\eta}^{+},\boldsymbol{y}^{t-1})] \text{ and}$$
$$P(\boldsymbol{y}^{-,t}|\boldsymbol{y}^{t-1};\boldsymbol{\eta}^{-}) = \exp[\boldsymbol{\eta}^{-}\cdot\boldsymbol{g}^{-}(\boldsymbol{y}^{-,t},\boldsymbol{y}^{t-1}) - \psi(\boldsymbol{\eta}^{-},\boldsymbol{y}^{t-1})].$$

In contrast to the TERGM whose parameter simultaneously influences both incidence and duration, STERGM provides two sets of parameters, where one manages ties formation and the other manages ties dissolution.

The intuition behind the separable parameterization is that the factors and processes that result in ties formation are different from those that result in ties dissolution. Many applications of STERGM on real-world data support the separable assumption for dynamic networks [BB18, ZDP19, XBS20, UH20, DYP21]. Despite the restriction that the two processes do not interact with each other, substantial improvement in interpretability is gained.

4.3 PST ERGM for Dynamic Valued Networks

4.3.1 Increment and Decrement Networks

Although we can use a TERGM [HFX10] to fit dynamic valued networks where $\boldsymbol{y}^{t-1}, \boldsymbol{y}^{t} \in \mathbb{N}_0^{\mathbb{Y}}$, parameter interpretation for the network evolution may be difficult. Consider the respective edge sum and stability terms as follows:

$$\boldsymbol{g}(\boldsymbol{y}^{t},\boldsymbol{y}^{t-1}) = \sum_{(i,j)\in\mathbb{Y}} \boldsymbol{y}_{ij}^{t} \text{ and } \boldsymbol{g}(\boldsymbol{y}^{t},\boldsymbol{y}^{t-1}) = \sum_{(i,j)\in\mathbb{Y}} \mathbb{1}(|\boldsymbol{y}_{ij}^{t} - \boldsymbol{y}_{ij}^{t-1}| \leq r),$$

where r is a threshold for the absolute difference in dyad value between time $t-1$ and t. In general, a higher coefficient on the edge sum term would favor networks that increase their dyad values at time t, while a higher coefficient on the stability term would favor networks that do not change their dyad values at time t. In this example, a higher coefficient in the TERGM may lead to inconsistent dyad value movement from time $t-1$ to t. Hence, a careful decomposition of the dyad value transition is also needed.

Intuitively, as relational phenomena evolve over time, it is assumed the factors and underlying processes that increase relational strength are different from those that decrease relational strength. For example, in an intensive care unit of a hospital, the number of contacts between a doctor and a patient may increase as the doctor frequently treats the patient during the early stage of infection. The count may further escalate if the patient's symptoms worsen. In contrast, the doctor may reduce the number of contacts as the patient later acquires immunity to the disease. The count may further plummet as the doctor uses medical sensors to monitor the patient after the symptoms alleviate.

Inspired by [KH14], we can also design two intermediate networks to consider the dyad value movement, separately. Given two consecutive valued networks $\boldsymbol{y}^{t-1}$ and $\boldsymbol{y}^{t}$, we construct an increment network $\boldsymbol{y}^{+,t}$ and a decrement network $\boldsymbol{y}^{-,t}$ between time $t-1$ and t for a dyad (i,j) with a scaling factor $\beta = 0.5$ as follows:

$$\begin{aligned} \boldsymbol{y}_{ij}^{+,t} &= f^{+}(\boldsymbol{y}_{ij}^{t-1}, \boldsymbol{y}_{ij}^{t}) \coloneqq 0.5(\boldsymbol{y}_{ij}^{t-1} + \boldsymbol{y}_{ij}^{t}) + \beta|\boldsymbol{y}_{ij}^{t-1} - \boldsymbol{y}_{ij}^{t}| = \max(\boldsymbol{y}_{ij}^{t-1}, \boldsymbol{y}_{ij}^{t}) \text{ and} \\ \boldsymbol{y}_{ij}^{-,t} &= f^{-}(\boldsymbol{y}_{ij}^{t-1}, \boldsymbol{y}_{ij}^{t}) \coloneqq 0.5(\boldsymbol{y}_{ij}^{t-1} + \boldsymbol{y}_{ij}^{t}) - \beta|\boldsymbol{y}_{ij}^{t-1} - \boldsymbol{y}_{ij}^{t}| = \min(\boldsymbol{y}_{ij}^{t-1}, \boldsymbol{y}_{ij}^{t}). \end{aligned} \tag{4.2}$$

Note that these are also generalized formulations of the formation network and the dissolution network in [KH14], since binary networks are special cases of valued networks in terms of dyad values.

Similar to the formation and dissolution networks, the increment and decrement networks for a dyad (i,j) appear to result in the observed $\boldsymbol{y}_{ij}^{t-1}$ and $\boldsymbol{y}_{ij}^{t}$. Intrinsically, $\boldsymbol{y}_{ij}^{+,t}$ and $\boldsymbol{y}_{ij}^{-,t}$ return the values from the average $0.5(\boldsymbol{y}_{ij}^{t-1} + \boldsymbol{y}_{ij}^{t})$ tilting away by the absolute difference $|\boldsymbol{y}_{ij}^{t-1} - \boldsymbol{y}_{ij}^{t}|$ scaled by a factor β between two consecutive time points. In this work, we use $\beta = 0.5$ to not exaggerate or diminish the absolute difference between the two time points, and to remain on count valued networks as $\boldsymbol{y}^{+,t}, \boldsymbol{y}^{-,t} \in \mathbb{N}_0^{\mathbb{Y}}$. The resulting max and min operations have further implications in model interpretation described in Section 4.3.3. For $\beta \neq 0.5$ that leads to $\boldsymbol{y}^{+,t}, \boldsymbol{y}^{-,t} \in \mathbb{R}^{\mathbb{Y}}$, an extension of our framework with the Generalized ERGM [DC12a] may be allowed for future development.

In summary, $\boldsymbol{y}^{+,t}$ contains the unchanged dyad values from time $t-1$ to t and those that

increased at time t, while $\boldsymbol{y}^{-,t}$ contains the unchanged dyad values from time $t-1$ to t and those that decreased at time t. Furthermore, both of them preserve the momentum when the dyad value starts to change in the opposite direction. As the bolded segments shown in Figure 4.1, the momentum of the changes is delayed to the next interval for a model to digest the stimulus. Similar to [KH14], we substitute the sequence of T observed networks with the sequence of $2 \times (T-1)$ extracted networks that focus on dyad value movements, as augmented input data to a model. Alternatively, $\boldsymbol{y}^{+,t}$ and $\boldsymbol{y}^{-,t}$ can be considered as two latent networks that emphasize the transitions between time $t-1$ and t, instead of two snapshots of the observed networks $\boldsymbol{y}^{t-1}$ and $\boldsymbol{y}^{t}$ which give limited information about the dynamics.

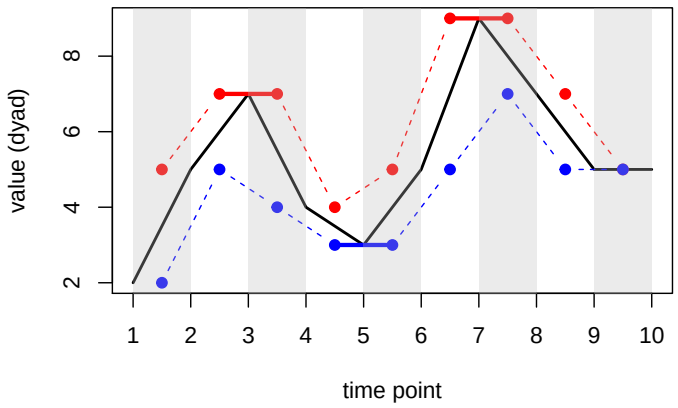

Figure 4.1: An illustration of $\boldsymbol{y}_{ij}^{t}$ (black) and the constructed $\boldsymbol{y}_{ij}^{+,t}$ (red) and $\boldsymbol{y}_{ij}^{-,t}$ (blue) over time.

Before proposing the PST ERGM in Section 4.3.2 with details, we further motivate the increment and decrement networks with a comparison between simple fitted models, using the baboons interaction networks [GGP20] analyzed in Section 4.5.3. As shown in Figure 4.2, the difference in edge sums $\sum_{(i,j)\in\mathbb{Y}} \boldsymbol{y}_{ij}^{t}$ between $t = 25$ and 26 is relatively small. Fitting a valued ERGM [Kri12a] that involves only the edge sum term to $\boldsymbol{y}^{26}$, we notice the coefficient is close to that of the same model fitted to $\boldsymbol{y}^{25}$. However, fitting the proposed PST ERGM that involves only the edge sum terms to both $\boldsymbol{y}^{+,26}$ and $\boldsymbol{y}^{-,26}$, we notice the coefficient of

$\sum_{(i,j)\in\mathbb{Y}} \boldsymbol{y}_{ij}^{+,26}$ is positive and that of $\sum_{(i,j)\in\mathbb{Y}} \boldsymbol{y}_{ij}^{-,26}$ is negative. The coefficients of the simple fitted models are displayed in 4.7.1.

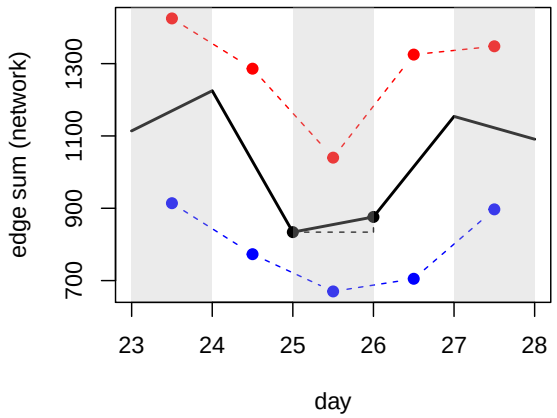

Figure 4.2: The edge sums of the baboons interaction networks $\boldsymbol{y}^t$ (black) from day 23 to 28, and the edge sums of the constructed $\boldsymbol{y}^{+,t}$ (red) and $\boldsymbol{y}^{-,t}$ (blue). The edge sums of $\boldsymbol{y}^{25}$ and $\boldsymbol{y}^{26}$ are highlighted.

In the example with PST ERGM, the positive coefficient indicates an increment among dyad values, while the negative coefficient indicates previously high dyad values tend not to persist to the next time point. There is a fluctuation in dyad values between the observed time points, though the edge sums appear to be unchanged at the observed time points. From $t = 25$ to 26, the total increment of the dyads that increase is 206, and the total decrement of the dyads that decrease is 164, resulting in a net increase of 42 or 11% of total value changes. Without a decomposition that separates dyad value increment and decrement, such dynamics may be neglected. Next, we introduce the proposed PST ERGM in details.

4.3.2 Model Specification

We first define the form of the model for a sequence of valued networks $\boldsymbol{y}^1, \cdots, \boldsymbol{y}^T$ with ERGM specified as the transition between consecutive networks. Under the first order

Markov assumption where $\boldsymbol{y}^t$ is independent of $\boldsymbol{y}^{t-2}, \cdots, \boldsymbol{y}^1$ conditioning on $\boldsymbol{y}^{t-1}$, we have

$$\begin{aligned} P(\boldsymbol{y}^T, \boldsymbol{y}^{T-1}, \cdots, \boldsymbol{y}^2 | \boldsymbol{y}^1) &= P(\boldsymbol{y}^T | \boldsymbol{y}^{T-1}) P(\boldsymbol{y}^{T-1} | \boldsymbol{y}^{T-2}) \cdots P(\boldsymbol{y}^2 | \boldsymbol{y}^1) \\ &= \prod_{t=2}^{T} h(\boldsymbol{y}^t, \boldsymbol{y}^{t-1}) \exp[\boldsymbol{\eta} \cdot \boldsymbol{g}(\boldsymbol{y}^t, \boldsymbol{y}^{t-1}) - \psi(\boldsymbol{\eta}, \boldsymbol{y}^{t-1})]. \end{aligned}$$

Besides the dynamics between consecutive networks, $P(\boldsymbol{y}^1)$ can be specified by a valued ERGM [Kri12a] to complete the joint distribution.

To dissect the entanglement between dyad value increment and decrement in dynamic valued networks, it may be straightforward to consider that the increment network $\boldsymbol{y}^{+,t}$ is also conditionally independent of the decrement network $\boldsymbol{y}^{-,t}$ given $\boldsymbol{y}^{t-1}$, as in the STERGM for dynamic binary networks. However, as we will compare the two cases below, a fully separable model for dynamic valued networks can be difficult to obtain, while retaining the information encoded in $\boldsymbol{y}^{+,t}$ and $\boldsymbol{y}^{-,t}$.

For dynamic binary networks where $\mathcal{Y}^t \subseteq 2^{\mathbb{Y}}$, the STERGM of (4.1) permits us to sample $\boldsymbol{y}^{+,t}$ and $\boldsymbol{y}^{-,t}$ individually to produce a unique $\boldsymbol{y}^t \in \mathcal{Y}^t$. Conditioning on $\boldsymbol{y}^{t-1}$ with a particular dyad $\boldsymbol{y}_{ij}^{t-1} = 0$, the sampled $\boldsymbol{y}_{ij}^{-,t}$ can only be 0 whereas the sampled $\boldsymbol{y}_{ij}^{+,t}$ can be either 0 or 1. Once $\boldsymbol{y}_{ij}^{+,t}$ is determined, a unique $\boldsymbol{y}_{ij}^t$ is confirmed. Similarly, conditioning on $\boldsymbol{y}^{t-1}$ with a particular dyad $\boldsymbol{y}_{ij}^{t-1} = 1$, the sampled $\boldsymbol{y}_{ij}^{+,t}$ can only be 1 whereas the sampled $\boldsymbol{y}_{ij}^{-,t}$ can be either 0 or 1. Once $\boldsymbol{y}_{ij}^{-,t}$ is determined, a unique $\boldsymbol{y}_{ij}^t$ is confirmed. Therefore, a separable model in the spaces of $\mathcal{Y}^{+,t}$ and $\mathcal{Y}^{-,t}$, proposed by [KH14], is a valid probability distribution for a binary network $\boldsymbol{y}^t$ conditional on $\boldsymbol{y}^{t-1}$.

For dynamic valued networks where $\mathcal{Y}^t \subseteq \mathbb{N}_0^{\mathbb{Y}}$, suppose $P(\boldsymbol{y}^t | \boldsymbol{y}^{t-1})$ can still be separated into two conditionally independent models as in (4.1), so that we can sample $\boldsymbol{y}^{+,t}$ and $\boldsymbol{y}^{-,t}$ individually to produce a unique $\boldsymbol{y}^t \in \mathcal{Y}^t$. Conditioning on $\boldsymbol{y}^{t-1}$ with a particular dyad value $\boldsymbol{y}_{ij}^{t-1} \in \mathbb{N}_0$ and under the specification of (4.2), a sampled $\boldsymbol{y}_{ij}^{+,t}$ can be any count value that is greater than or equal to $\boldsymbol{y}_{ij}^{t-1}$, and a sampled $\boldsymbol{y}_{ij}^{-,t}$ can be any non-negative count value that is smaller than or equal to $\boldsymbol{y}_{ij}^{t-1}$. For example, conditioning on $\boldsymbol{y}^{t-1}$ with $\boldsymbol{y}_{ij}^{t-1} = 3$, if the sampled $\boldsymbol{y}_{ij}^{+,t}$ from $P(\boldsymbol{y}^{+,t} | \boldsymbol{y}^{t-1}; \boldsymbol{\eta}^+)$ is 5 and the sampled $\boldsymbol{y}_{ij}^{-,t}$ from $P(\boldsymbol{y}^{-,t} | \boldsymbol{y}^{t-1}; \boldsymbol{\eta}^-)$

is 2, a unique $\boldsymbol{y}_{ij}^t$ is unidentifiable given the two intermediate dyad values. The separated generating processes cannot decide whether the dyad value $\boldsymbol{y}_{ij}^{t-1} = 3$ should increase to $\boldsymbol{y}_{ij}^t = 5$ or decrease to $\boldsymbol{y}_{ij}^t = 2$ at time t.

Since the TERGM in our framework can no longer be separated into two conditionally independent models as in [KH14], our proposed Partially Separable Temporal ERGM (PST ERGM) for a sequence of valued networks is

$$\begin{aligned}\prod_{t=2}^{T} P(\boldsymbol{y}^t|\boldsymbol{y}^{t-1};\boldsymbol{\eta}) &= \prod_{t=2}^{T} h(\boldsymbol{y}^t,\boldsymbol{y}^{t-1})\exp[\boldsymbol{\eta}\cdot\boldsymbol{g}(\boldsymbol{y}^t,\boldsymbol{y}^{t-1}) - \psi(\boldsymbol{\eta},\boldsymbol{y}^{t-1})] \\ &= \prod_{t=2}^{T} h^+(\boldsymbol{y}^{+,t})h^-(\boldsymbol{y}^{-,t})\frac{\exp[\boldsymbol{\eta}^+\cdot\boldsymbol{g}^+(\boldsymbol{y}^{+,t},\boldsymbol{y}^{t-1}) + \boldsymbol{\eta}^-\cdot\boldsymbol{g}^-(\boldsymbol{y}^{-,t},\boldsymbol{y}^{t-1})]}{\exp[\psi(\boldsymbol{\eta}^+,\boldsymbol{\eta}^-,\boldsymbol{y}^{t-1})]},\end{aligned} \tag{4.3}$$

with $h(\boldsymbol{y}^t,\boldsymbol{y}^{t-1}) = h^+(\boldsymbol{y}^{+,t}) \times h^-(\boldsymbol{y}^{-,t}) \in \mathbb{R}$ and $\boldsymbol{\eta} = (\boldsymbol{\eta}^+,\boldsymbol{\eta}^-) \in \mathbb{R}^p$. The network statistics $\boldsymbol{g}(\boldsymbol{y}^t,\boldsymbol{y}^{t-1}) = \big(\boldsymbol{g}^+(\boldsymbol{y}^{+,t},\boldsymbol{y}^{t-1}),\boldsymbol{g}^-(\boldsymbol{y}^{-,t},\boldsymbol{y}^{t-1})\big) \in \mathbb{R}^p$ is a concatenation of the increment network statistics $\boldsymbol{g}^+(\boldsymbol{y}^{+,t},\boldsymbol{y}^{t-1}) \in \mathbb{R}^{p_1}$ and the decrement network statistics $\boldsymbol{g}^-(\boldsymbol{y}^{-,t},\boldsymbol{y}^{t-1}) \in \mathbb{R}^{p_2}$ such that $p_1 + p_2 = p$. The normalizing constant $\exp[\psi(\boldsymbol{\eta}^+,\boldsymbol{\eta}^-,\boldsymbol{y}^{t-1})]$ is

$$\sum_{\boldsymbol{y}^t\in\mathcal{Y}^t} h^+(\boldsymbol{y}^{+,t})\exp[\boldsymbol{\eta}^+\cdot\boldsymbol{g}^+(\boldsymbol{y}^{+,t},\boldsymbol{y}^{t-1})] \times h^-(\boldsymbol{y}^{-,t})\exp[\boldsymbol{\eta}^-\cdot\boldsymbol{g}^-(\boldsymbol{y}^{-,t},\boldsymbol{y}^{t-1})].$$

Though we cannot generate $\boldsymbol{y}^{+,t}$ and $\boldsymbol{y}^{-,t}$ to produce a unique $\boldsymbol{y}^t$ with PST ERGM, we can directly sample $\boldsymbol{y}^t$ by using the Metropolis-Hastings algorithm described in Section 4.4.3.

In this chapter, we use the Poisson and Binomial reference functions:

$$h^+(\boldsymbol{y}^{+,t}) = \prod_{(i,j)\in\mathbb{Y}} (\boldsymbol{y}_{ij}^{+,t}!)^{-1} \quad \text{and} \quad h^-(\boldsymbol{y}^{-,t}) = \prod_{(i,j)\in\mathbb{Y}} \binom{m^t}{\boldsymbol{y}_{ij}^{-,t}}, \tag{4.4}$$

for the increment and decrement process, respectively. The term m^t in $h^-(\boldsymbol{y}^{-,t})$ is a predetermined maximum value that each dyad value $\boldsymbol{y}_{ij}^{-,t} \in \{0,1,\cdots,m^t\}$ can take in $\boldsymbol{y}^{-,t}$. Furthermore, the reference function $h^+(\boldsymbol{y}^{+,t})$ in the increment process does not require an upper bound for a dyad value $\boldsymbol{y}_{ij}^{+,t}$ that it can increase to, but $\boldsymbol{y}_{ij}^{+,t}$ has an implicit lower bound that is equal to $\boldsymbol{y}_{ij}^{t-1}$ inherited from the construction of $\boldsymbol{y}_{ij}^{+,t} = \max(\boldsymbol{y}_{ij}^{t-1},\boldsymbol{y}_{ij}^t)$. In

the decrement process, the reference function $h^-(\boldsymbol{y}^{-,t})$ imposes an upper bound m^t for each dyad value $\boldsymbol{y}_{ij}^{-,t}$ that it can decrease from, with an explicit lower bound that is equal to 0.

To capture the variation in structural properties between different intervals, we can also specify a time-heterogeneous PST ERGM $\prod_{t=2}^{T} P(\boldsymbol{y}^t|\boldsymbol{y}^{t-1};\boldsymbol{\eta}^t)$ as

$$\prod_{t=2}^{T} \frac{h^+(\boldsymbol{y}^{+,t})\exp[\boldsymbol{\eta}^{+,t}\cdot\boldsymbol{g}^+(\boldsymbol{y}^{+,t},\boldsymbol{y}^{t-1})]\times h^-(\boldsymbol{y}^{-,t})\exp[\boldsymbol{\eta}^{-,t}\cdot\boldsymbol{g}^-(\boldsymbol{y}^{-,t},\boldsymbol{y}^{t-1})]}{\exp[\psi(\boldsymbol{\eta}^{+,t},\boldsymbol{\eta}^{-,t},\boldsymbol{y}^{t-1})]},$$

where $\boldsymbol{\eta}^t = (\boldsymbol{\eta}^{+,t},\boldsymbol{\eta}^{-,t})$ differs by time t. Unless otherwise noted, we focus on the time-homogeneous PST ERGM of (4.3) whose parameter $\boldsymbol{\eta} = (\boldsymbol{\eta}^+,\boldsymbol{\eta}^-)$ is fixed across $t = 2,\cdots,T$. The time-heterogeneous PST ERGM is a special case of (4.3) as $\boldsymbol{\eta}^t = (\boldsymbol{\eta}^{+,t},\boldsymbol{\eta}^{-,t})$ can be learned sequentially for each t.

4.3.2.1 Reference Measures

Inherited from valued ERGM [Kri12a], the reference function $h(\boldsymbol{y}^t,\boldsymbol{y}^{t-1})$ specified with (4.4) in a PST ERGM underlies a baseline distribution for $\boldsymbol{y}^t$. Consider a PST ERGM with the edge sums of increment and decrement networks as two network statistics:

$$\boldsymbol{g}(\boldsymbol{y}^t,\boldsymbol{y}^{t-1}) = \big(\boldsymbol{g}^+(\boldsymbol{y}^{+,t},\boldsymbol{y}^{t-1}),\boldsymbol{g}^-(\boldsymbol{y}^{-,t},\boldsymbol{y}^{t-1})\big) = \Big(\sum_{(i,j)\in\mathbb{Y}} \boldsymbol{y}_{ij}^{+,t}, \sum_{(i,j)\in\mathbb{Y}} \boldsymbol{y}_{ij}^{-,t}\Big) \in \mathbb{R}^2.$$

Let $\mathcal{Y}^t(\boldsymbol{y}^{t-1}) \subseteq \{\boldsymbol{y}^t \in \mathbb{N}_0^{\mathbb{Y}} : \boldsymbol{y}_{ij}^t > \boldsymbol{y}_{ij}^{t-1}\ \forall\ (i,j)\in\mathbb{Y}\}$ be a sample space for $\boldsymbol{y}^t$ starting from $\boldsymbol{y}^{t-1}$. The increment network $\boldsymbol{y}^{+,t}$ is essentially the $\boldsymbol{y}^t$, and the PST ERGM $P(\boldsymbol{y}^t|\boldsymbol{y}^{t-1};\boldsymbol{\eta})$ for $\boldsymbol{y}^t \in \mathcal{Y}^t(\boldsymbol{y}^{t-1})$ becomes a dyadic independent truncated Poisson distribution:

$$\prod_{(i,j)\in\mathbb{Y}} \frac{(\boldsymbol{y}_{ij}^t!)^{-1}\exp(\boldsymbol{\eta}^+\cdot\boldsymbol{y}_{ij}^t)}{\exp\big(\exp(\boldsymbol{\eta}^+)\big) - \sum_{u=0}^{\boldsymbol{y}_{ij}^{t-1}}(u!)^{-1}\exp(\boldsymbol{\eta}^+\cdot u)} = \prod_{(i,j)\in\mathbb{Y}} \frac{P_{\text{Pois}}\big(\boldsymbol{y}_{ij}^t\big)}{1-\sum_{u=0}^{\boldsymbol{y}_{ij}^{t-1}} P_{\text{Pois}}(u)},$$

where $P_{\text{Pois}}(x)$ denotes the probability mass function of Poisson$\big(\lambda = \exp(\boldsymbol{\eta}^+)\big)$ evaluated at x. Moreover, let $\mathcal{Y}^t(\boldsymbol{y}^{t-1}) \subseteq \{\boldsymbol{y}^t \in \mathbb{N}_0^{\mathbb{Y}} : \boldsymbol{y}_{ij}^t < \boldsymbol{y}_{ij}^{t-1} \le m^t\ \forall\ (i,j)\in\mathbb{Y}\}$ be another sample space for $\boldsymbol{y}^t$. The decrement network $\boldsymbol{y}^{-,t}$ is essentially the $\boldsymbol{y}^t$, and the PST ERGM $P(\boldsymbol{y}^t|\boldsymbol{y}^{t-1};\boldsymbol{\eta})$

for $\boldsymbol{y}^t \in \mathcal{Y}^t(\boldsymbol{y}^{t-1})$ becomes a dyadic independent truncated Binomial distribution:

$$\prod_{(i,j)\in\mathbb{Y}} \frac{\binom{m^t}{\boldsymbol{y}_{ij}^t}\exp(\boldsymbol{\eta}^- \cdot \boldsymbol{y}_{ij}^t)}{\big(1+\exp(\boldsymbol{\eta}^-)\big)^{m^t} - \sum_{u=\boldsymbol{y}_{ij}^{t-1}}^{m^t}\binom{m^t}{u}\exp(\boldsymbol{\eta}^- \cdot u)} = \prod_{(i,j)\in\mathbb{Y}} \frac{P_{\text{Bino}}\big(\boldsymbol{y}_{ij}^t\big)}{1-\sum_{u=\boldsymbol{y}_{ij}^{t-1}}^{m^t} P_{\text{Bino}}\big(u\big)},$$

where $P_{\text{Bino}}(x)$ denotes the probability mass function of Binomial$\big(m^t, p = \text{logit}^{-1}(\boldsymbol{\eta}^-)\big)$ evaluated at x. The derivations of the two special cases are provided in 4.7.2.

Next, we consider the general network statistics $\boldsymbol{g}(\boldsymbol{y}^t, \boldsymbol{y}^{t-1}) \in \mathbb{R}^p$. For $\mathcal{Y}^t(\boldsymbol{y}^{t-1}) \subseteq \{\boldsymbol{y}^t \in \mathbb{N}_0^{\mathbb{Y}} : \boldsymbol{y}_{ij}^t > \boldsymbol{y}_{ij}^{t-1}\ \forall\ (i,j) \in \mathbb{Y}\}$, the PST ERGM becomes a Poisson-reference valued ERGM:

$$P(\boldsymbol{y}^t|\boldsymbol{y}^{t-1};\boldsymbol{\eta}) = \frac{\big(\prod_{(i,j)\in\mathbb{Y}}(\boldsymbol{y}_{ij}^t!)^{-1}\big)\exp[\boldsymbol{\eta}^+ \cdot \boldsymbol{g}^+(\boldsymbol{y}^t, \boldsymbol{y}^{t-1})]}{\sum_{\boldsymbol{y}^t\in\mathcal{Y}^t(\boldsymbol{y}^{t-1})}\big(\prod_{(i,j)\in\mathbb{Y}}(\boldsymbol{y}_{ij}^t!)^{-1}\big)\exp[\boldsymbol{\eta}^+ \cdot \boldsymbol{g}^+(\boldsymbol{y}^t, \boldsymbol{y}^{t-1})]},$$

with the Poisson reference function and the increment network statistics directly evaluated at $\boldsymbol{y}^t \in \mathcal{Y}^t(\boldsymbol{y}^{t-1})$. Moreover, for $\mathcal{Y}^t(\boldsymbol{y}^{t-1}) \subseteq \{\boldsymbol{y}^t \in \mathbb{N}_0^{\mathbb{Y}} : \boldsymbol{y}_{ij}^t < \boldsymbol{y}_{ij}^{t-1} \le m^t\ \forall\ (i,j) \in \mathbb{Y}\}$, the PST ERGM becomes a Binomial-reference valued ERGM:

$$P(\boldsymbol{y}^t|\boldsymbol{y}^{t-1};\boldsymbol{\eta}) = \frac{\big(\prod_{(i,j)\in\mathbb{Y}}\binom{m^t}{\boldsymbol{y}_{ij}^t}\big)\exp[\boldsymbol{\eta}^- \cdot \boldsymbol{g}^-(\boldsymbol{y}^t, \boldsymbol{y}^{t-1})]}{\sum_{\boldsymbol{y}^t\in\mathcal{Y}^t(\boldsymbol{y}^{t-1})}\big(\prod_{(i,j)\in\mathbb{Y}}\binom{m^t}{\boldsymbol{y}_{ij}^t}\big)\exp[\boldsymbol{\eta}^- \cdot \boldsymbol{g}^-(\boldsymbol{y}^t, \boldsymbol{y}^{t-1})]},$$

with the Binomial reference function and the decrement network statistics directly evaluated at $\boldsymbol{y}^t \in \mathcal{Y}^t(\boldsymbol{y}^{t-1})$. Next, we provide four levels of interpretation to PST ERGM.

4.3.3 Model Interpretation

We first compare the formulations of STERGM and the proposed PST ERGM. The STERGM [KH14] for dynamic binary networks given as

$$P(\boldsymbol{y}^t|\boldsymbol{y}^{t-1};\boldsymbol{\eta}) = \frac{\exp[\boldsymbol{\eta}^+ \cdot \boldsymbol{g}^+(\boldsymbol{y}^{+,t}, \boldsymbol{y}^{t-1})]}{\exp[\psi(\boldsymbol{\eta}^+, \boldsymbol{y}^{t-1})]} \times \frac{\exp[\boldsymbol{\eta}^- \cdot \boldsymbol{g}^-(\boldsymbol{y}^{-,t}, \boldsymbol{y}^{t-1})]}{\exp[\psi(\boldsymbol{\eta}^-, \boldsymbol{y}^{t-1})]}$$

is fully separable, while the PST ERGM for dynamic valued networks given as

$$P(\boldsymbol{y}^t|\boldsymbol{y}^{t-1};\boldsymbol{\eta}) \propto\ \exp[\boldsymbol{\eta}^+ \cdot \boldsymbol{g}^+(\boldsymbol{y}^{+,t}, \boldsymbol{y}^{t-1})] \times \exp[\boldsymbol{\eta}^- \cdot \boldsymbol{g}^-(\boldsymbol{y}^{-,t}, \boldsymbol{y}^{t-1})]$$

is partially separable. Though both models are distributions over the observed networks, the user-specified network statistics $\boldsymbol{g}^+$ and $\boldsymbol{g}^-$ in both models are evaluated at $\boldsymbol{y}^{+,t}$ and $\boldsymbol{y}^{-,t}$ that

are extracted from the observed networks. Hierarchically, two layers of network features are extracted: dyad value movements via $\boldsymbol{y}^{+,t}$ and $\boldsymbol{y}^{-,t}$, and their structural properties via $\boldsymbol{g}^{+}$ and $\boldsymbol{g}^{-}$. These dynamics are then captured by $\boldsymbol{g}(\boldsymbol{y}^{t}, \boldsymbol{y}^{t-1}) = \big(\boldsymbol{g}^{+}(\boldsymbol{y}^{+,t}, \boldsymbol{y}^{t-1}), \boldsymbol{g}^{-}(\boldsymbol{y}^{-,t}, \boldsymbol{y}^{t-1})\big)$ with an exponential-family model. In alignment with the separability, the choice of network statistics in $\boldsymbol{g}^{+}(\boldsymbol{y}^{+,t}, \boldsymbol{y}^{t-1}) \in \mathbb{R}^{p_1}$ can be different from that in $\boldsymbol{g}^{-}(\boldsymbol{y}^{-,t}, \boldsymbol{y}^{t-1}) \in \mathbb{R}^{p_2}$, depending on the user's knowledge of which local forces matter in which process to shape the global structures over time. In contrast to STERGM, the increment and decrement processes in the PST ERGM are not conditionally independent, as discussed in Section 4.3.2.

Next, we present the similarity between STERGM and PST ERGM in how they dissect network evolution. The idea of separability originates from epidemiology to approximate disease dynamics: Prevalence $\approx$ Incidence $\times$ Duration. [KH14] used the formation and dissolution networks to reflect incidence (how often new ties are formed) and duration (how long old ties last since they were formed). Since the duration of ties is the inverse of the rate at which ties dissolve, the parameter $\boldsymbol{\eta}^{-}$ in the dissolution model of STERGM can signify the persistence of network features [KH14]. Translating these into PST ERGM, the structural properties of dynamic valued networks are characterizations of the amount and rate of dyad value increment and decrement. The more often dyad values increase and they increase with a greater magnitude per time step, the more high dyad values will be presented over time. The less frequent dyad values decrease and they decrease with a smaller magnitude per time step, the more high dyad values will be preserved over time. Conceptually, we can broadly regard incidence as how often dyad values increase, and regard duration as how long dyad values have kept increasing until they decrease. The duration of the continuing increment is the inverse of the rate at which dyad values decrease. Inherited from Equation (4.2), the increment and decrement networks are also encoded with the values to which the dyads have increased or decreased.

Furthermore, the ERGM framework has a dyadic level interpretation for a static binary network in terms of change statistics [HHB08b]. We provide similar interpretation for

dynamic valued networks with PST ERGM, by changing a dyad value in $\boldsymbol{y}^t$ to see how increment and decrement processes impact the network structures. Specifically, we calculate the ratio of probabilities of two networks that are identical except for a single dyad. Suppose the dyad value $\boldsymbol{y}_{ij}^t \in \mathbb{N}_0$ jumps from a to b where $b \neq a$. Conditioning on the rest of the network $\boldsymbol{y}_{-ij}^t$ and $\boldsymbol{y}^{t-1}$, the ratio $P(\boldsymbol{y}_{ij}^t = b|\boldsymbol{y}_{-ij}^t, \boldsymbol{y}^{t-1})/P(\boldsymbol{y}_{ij}^t = a|\boldsymbol{y}_{-ij}^t, \boldsymbol{y}^{t-1})$ is

$$\frac{\boldsymbol{y}_{\text{old}}^{+,t}!}{\boldsymbol{y}_{\text{new}}^{+,t}!} \exp[\boldsymbol{\eta}^+ \cdot \Delta \boldsymbol{g}^+(\boldsymbol{y}^{+,t}, \boldsymbol{y}^{t-1})_{ij}] \times \binom{m^t}{\boldsymbol{y}_{\text{new}}^{-,t}} / \binom{m^t}{\boldsymbol{y}_{\text{old}}^{-,t}} \exp[\boldsymbol{\eta}^- \cdot \Delta \boldsymbol{g}^-(\boldsymbol{y}^{-,t}, \boldsymbol{y}^{t-1})_{ij}].$$

The change statistics $\Delta \boldsymbol{g}^+(\boldsymbol{y}^{+,t}, \boldsymbol{y}^{t-1})_{ij}$ denote the difference between $\boldsymbol{g}^+(\boldsymbol{y}^{+,t}, \boldsymbol{y}^{t-1})$ with $\boldsymbol{y}_{ij}^{+,t} = \boldsymbol{y}_{\text{new}}^{+,t}$ and $\boldsymbol{g}^+(\boldsymbol{y}^{+,t}, \boldsymbol{y}^{t-1})$ with $\boldsymbol{y}_{ij}^{+,t} = \boldsymbol{y}_{\text{old}}^{+,t}$ while rest of the $\boldsymbol{y}^{+,t}$ remains the same, and the $\Delta \boldsymbol{g}^-(\boldsymbol{y}^{-,t}, \boldsymbol{y}^{t-1})_{ij}$ are denoted similarly except for notational difference. When both $a, b > \boldsymbol{y}_{ij}^{t-1}$, only $\boldsymbol{y}^{+,t}$ by construction is updated, regardless of $a > b$ or $b > a$. In other words, only the increment process contributes to the structural changes in $\boldsymbol{y}^t$ when the dyad value $\boldsymbol{y}_{ij}^t$ is different. Similarly, when both $a, b < \boldsymbol{y}_{ij}^{t-1}$, only the decrement process contributes to the structural changes in $\boldsymbol{y}^t$. However, when the value b falls on the other side of $\boldsymbol{y}_{ij}^{t-1}$ with respect to the value a, both increment and decrement processes start to contribute to the structural changes in $\boldsymbol{y}^t$. Intuitively, the construction by (4.2) can be considered as rectified linear units, gated by $\boldsymbol{y}_{ij}^{t-1}$ that differs by dyad (i, j) and time point t. When a dyad value overcomes a threshold, it activates the corresponding process via user-specified network statistics to impact the network structures.

In addition to the dyadic level interpretation, the parameters of PST ERGM can be interpreted at the structural level, as in [KH14]. Though we cannot generate $\boldsymbol{y}^{+,t}$ and $\boldsymbol{y}^{-,t}$ to produce $\boldsymbol{y}^t$ given fixed parameters, we can learn the unknown parameters of PST ERGM given observed $\boldsymbol{y}^t$ and $\boldsymbol{y}^{t-1}$ to interpret the dynamics via $\boldsymbol{g}(\boldsymbol{y}^t, \boldsymbol{y}^{t-1})$ from an exponential-family model. When fitting a PST ERGM on the observed $\boldsymbol{y}^t$ conditional on $\boldsymbol{y}^{t-1}$, the dyad value movements (increased, decreased, or unchanged) between consecutive time points become fixed. As we augment the observed networks with a sequence of $2 \times (T-1)$ networks that separate dyad value increment and decrement, the learned parameters $(\boldsymbol{\eta}^+, \boldsymbol{\eta}^-) \in \mathbb{R}^p$ can signify the structural changes in $\boldsymbol{y}^t$ stemming from the two processes. In general, for a

particular positive $\boldsymbol{g}_i^+(\boldsymbol{y}^{+,t}, \boldsymbol{y}^{t-1})$ in the increment process, a positive $\boldsymbol{\eta}_i^+$ is associated with increasing dyad values to have more instances of the feature that is tracked by $\boldsymbol{g}_i^+(\boldsymbol{y}^{+,t}, \boldsymbol{y}^{t-1})$ in the extracted $\boldsymbol{y}^{+,t}$. On the contrary, a negative $\boldsymbol{\eta}_i^+$ will disrupt the emergence of this feature by not increasing dyad values, resulting in fewer instances of the feature in $\boldsymbol{y}^{+,t}$. For a particular positive $\boldsymbol{g}_i^-(\boldsymbol{y}^{-,t}, \boldsymbol{y}^{t-1})$ in the decrement process, a positive $\boldsymbol{\eta}_i^-$ is associated with not decreasing dyad values to have more instances of the feature that is tracked by $\boldsymbol{g}_i^-(\boldsymbol{y}^{-,t}, \boldsymbol{y}^{t-1})$ in the extracted $\boldsymbol{y}^{-,t}$. However, a negative $\boldsymbol{\eta}_i^-$ will target this feature by reducing dyad values, resulting in fewer instances of the feature in $\boldsymbol{y}^{-,t}$. Equivalently, a negative $\boldsymbol{\eta}_i^-$ is associated with a shorter duration of the feature appearance. Moreover, since we learn the parameters $(\boldsymbol{\eta}^+, \boldsymbol{\eta}^-)$ jointly as described in Section 4.4, the parameters that reflect the dynamics via $\boldsymbol{g}(\boldsymbol{y}^t, \boldsymbol{y}^{t-1}) = \big(\boldsymbol{g}^+(\boldsymbol{y}^{+,t}, \boldsymbol{y}^{t-1}), \boldsymbol{g}^-(\boldsymbol{y}^{-,t}, \boldsymbol{y}^{t-1})\big)$ balance the two processes. Though we assume the factors that increase relational strength are different from those that decrease relational strength, they can be interacting in practice and the effects of interactions over time are absorbed into $(\boldsymbol{\eta}^+, \boldsymbol{\eta}^-)$ with a partially separable model.

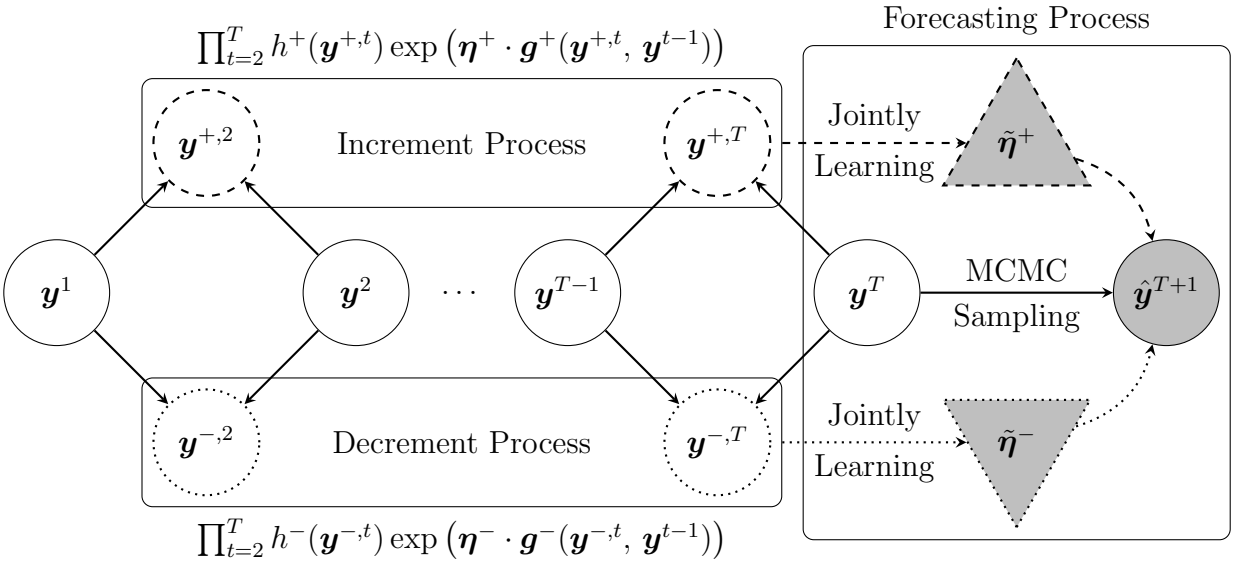

Figure 4.3: An overview of PST ERGM for dynamic valued networks.

Figure 4.3 gives an overview of the PST ERGM framework. The white solid circles denote

the sequence of observed networks as time passes from left to right. The dashed circles denote the sequence of increment networks, and the dotted circles denote the sequence of decrement networks. Note that each observed network is utilized multiple times to extract information that emphasizes the transition between consecutive time steps. The model with respect to the observed networks is partially separated into the increment process and the decrement process. Once the parameters in the two triangles are learned jointly, we can perform MCMC sampling to generate $\hat{\boldsymbol{y}}^{T+1}$ in the forecasting process. Though $\boldsymbol{y}^t$ can be further conditioned on more previous networks to calculate the network statistics and to construct $\boldsymbol{y}^{+,t}$ and $\boldsymbol{y}^{-,t}$, we only discuss PST ERGM under first order Markov assumption in this chapter.

4.4 Likelihood-Based Inference

The PST ERGM parameter estimation consists of two phases, extended from recent advances in fitting static binary networks. Specifically, we first maximize the log-likelihood ratio to seed an initial configuration, followed by the Newton-Raphson method to refine the parameters. The algorithms are provided in 4.7.3.

4.4.1 Log-likelihood Ratio

Throughout, the parameters $(\boldsymbol{\eta}^+, \boldsymbol{\eta}^-) = \boldsymbol{\eta} \in \mathbb{R}^p$ are estimated jointly. The log-likelihood of PST ERGM in (4.3) with $\big(\boldsymbol{g}^+(\boldsymbol{y}^{+,t}, \boldsymbol{y}^{t-1}), \boldsymbol{g}^-(\boldsymbol{y}^{-,t}, \boldsymbol{y}^{t-1})\big) = \boldsymbol{g}(\boldsymbol{y}^t, \boldsymbol{y}^{t-1}) \in \mathbb{R}^p$ is

$$l(\boldsymbol{\eta}) = \sum_{t=2}^{T} \Big\{ \log[h(\boldsymbol{y}^t, \boldsymbol{y}^{t-1})] + \boldsymbol{\eta} \cdot \boldsymbol{g}(\boldsymbol{y}^t, \boldsymbol{y}^{t-1}) - \psi(\boldsymbol{\eta}, \boldsymbol{y}^{t-1}) \Big\}.$$

The term $\exp[\psi(\boldsymbol{\eta}, \boldsymbol{y}^{t-1})]$ involves a sum over all possible networks in $\mathcal{Y}^t \subseteq \mathbb{N}_0^{\mathbb{Y}}$, which is often computationally intractable except for models with particular conditional independence properties [LRS18] or small networks [YSH21]. Consequently, we approximate the MLE using MCMC methods. To maximize the log-likelihood, we calculate its first and second derivative

with respect to $\boldsymbol{\eta}$:

$$\boldsymbol{S}(\boldsymbol{\eta}) = \sum_{t=2}^{T} \Big\{ \boldsymbol{g}(\boldsymbol{y}^t, \boldsymbol{y}^{t-1}) - \mathbb{E}_{\boldsymbol{\eta}}[\boldsymbol{g}(\boldsymbol{y}^t, \boldsymbol{y}^{t-1})] \Big\} \quad \text{and} \quad \boldsymbol{H}(\boldsymbol{\eta}) = \sum_{t=2}^{T} -\text{Cov}[\boldsymbol{g}(\boldsymbol{y}^t, \boldsymbol{y}^{t-1})]. \tag{4.5}$$

The gradient $\boldsymbol{S}(\boldsymbol{\eta})$ illustrates that ERGM fitting is essentially a feature pursuit: finding a parameter $\boldsymbol{\eta}$ such that the expected network statistics are close to the observed network statistics. Moreover, to obtain the standard errors of $\boldsymbol{\eta}$, the Fisher Information matrix can be approximated by the Hessian as $\boldsymbol{I}(\boldsymbol{\eta}) \approx -\hat{\boldsymbol{H}}(\tilde{\boldsymbol{\eta}})$ evaluated at the learned parameter $\tilde{\boldsymbol{\eta}}$ with MCMC samples [HH06b].

Approximating $\mathbb{E}_{\boldsymbol{\eta}}[\boldsymbol{g}(\boldsymbol{y}^t, \boldsymbol{y}^{t-1})]$ and $\text{Cov}[\boldsymbol{g}(\boldsymbol{y}^t, \boldsymbol{y}^{t-1})]$ of (4.5) with MCMC samples, the parameter $\boldsymbol{\eta}$ can be updated iteratively by the Newton-Raphson method. However, generating new MCMC samples at each learning iteration is computationally expensive. To reduce the computational burden, we use the log-likelihood ratio as a new objective function to approximate the MLE as in [Sni02], and [HH06b]. Let $\boldsymbol{\eta}_0$ be another initialized parameter, the log-likelihood ratio is

$$r(\boldsymbol{\eta}, \boldsymbol{\eta}_0) = \sum_{t=2}^{T} \Big\{ (\boldsymbol{\eta} - \boldsymbol{\eta}_0)^\top \boldsymbol{g}(\boldsymbol{y}^t, \boldsymbol{y}^{t-1}) - \log \mathbb{E}_{\boldsymbol{\eta}_0} \big(\exp[(\boldsymbol{\eta} - \boldsymbol{\eta}_0)^\top \boldsymbol{g}(\boldsymbol{y}^t, \boldsymbol{y}^{t-1})] \big) \Big\}.$$

Note that the distribution to draw samples from is now changed as we introduce an initialized parameter $\boldsymbol{\eta}_0$ to the log-likelihood ratio.

Generating a sufficiently large number of samples from $P(\boldsymbol{y}^t | \boldsymbol{y}^{t-1}; \boldsymbol{\eta}_0)$ only once for each time t, and then iterating a Newton-Raphson method with respect to $\boldsymbol{\eta}$ until convergence yields a maximizer of the approximated log-likelihood ratio. Though anchored on predetermined samples can greatly expedite the estimation, the efficiency of not having to update the MCMC samples between learning iterations comes with a cost. [GT92] pointed out that the approximated log-likelihood ratio via MCMC samples degrades quickly as $\boldsymbol{\eta}_0$ moves away from $\boldsymbol{\eta}$. We address this issue in the next section.

4.4.2 Normality Approximation and Partial Stepping

[HHH12b] proposed two amendments to improve the fitting for static binary networks. We adapt them to the PST ERGM for dynamic valued networks, to seed a starting point for the Newton-Raphson method. Let $\boldsymbol{y}_1^t, \cdots, \boldsymbol{y}_s^t$ be a list of s networks sampled from $P(\boldsymbol{y}^t|\boldsymbol{y}^{t-1}; \boldsymbol{\eta}_0)$. Since they are drawn from the same distribution, we can assume their network statistics multiplied by the difference of the two parameters, $(\boldsymbol{\eta} - \boldsymbol{\eta}_0)^\top \boldsymbol{g}(\boldsymbol{y}^t, \boldsymbol{y}^{t-1})$, follow a normal distribution $\mathcal{N}(\mu_t, \sigma_t^2)$ with

$$\mu_t = (\boldsymbol{\eta} - \boldsymbol{\eta}_0)^\top \boldsymbol{\mu}_t \quad \text{and} \quad \sigma_t^2 = (\boldsymbol{\eta} - \boldsymbol{\eta}_0)^\top \boldsymbol{\Sigma}_t (\boldsymbol{\eta} - \boldsymbol{\eta}_0).$$

The $\boldsymbol{\mu}_t$ and $\boldsymbol{\Sigma}_t$ are the respective mean vector and covariance matrix of $\boldsymbol{g}(\boldsymbol{y}^t, \boldsymbol{y}^{t-1})$ evaluated from the sampled $\boldsymbol{y}_1^t, \cdots, \boldsymbol{y}_s^t$. Given that $\exp[(\boldsymbol{\eta} - \boldsymbol{\eta}_0)^\top \boldsymbol{g}(\boldsymbol{y}^t, \boldsymbol{y}^{t-1})]$ is now log-normally distributed, the ratio of the two normalizing constants at t can be replaced by

$$\mathbb{E}_{\boldsymbol{\eta}_0}\big(\exp[(\boldsymbol{\eta} - \boldsymbol{\eta}_0)^\top \boldsymbol{g}(\boldsymbol{y}^t, \boldsymbol{y}^{t-1})]\big) \approx \exp(\mu_t + \frac{1}{2}\sigma_t^2),$$

and the approximated log-likelihood ratio becomes

$$\hat{r}_s(\boldsymbol{\eta}, \boldsymbol{\eta}_0) = (\boldsymbol{\eta} - \boldsymbol{\eta}_0)^\top \Big[\sum_{t=2}^{T} \boldsymbol{g}(\boldsymbol{y}^t, \boldsymbol{y}^{t-1}) - \sum_{t=2}^{T} \boldsymbol{\mu}_t\Big] - \frac{1}{2}(\boldsymbol{\eta} - \boldsymbol{\eta}_0)^\top \Big[\sum_{t=2}^{T} \boldsymbol{\Sigma}_t\Big](\boldsymbol{\eta} - \boldsymbol{\eta}_0).$$

Although the approximated log-likelihood ratio $\hat{r}_s(\boldsymbol{\eta}, \boldsymbol{\eta}_0)$ degrades quickly as $\boldsymbol{\eta}_0$ moves away from $\boldsymbol{\eta}$, we can restrict the amount of parameter update to prevent the degradation of $\hat{r}_s(\boldsymbol{\eta}, \boldsymbol{\eta}_0)$. With a step length $\gamma^t \in (0, 1]$ at time t, we create a pseudo-observation

$$\hat{\boldsymbol{\xi}}(\boldsymbol{y}) = \sum_{t=2}^{T} \gamma^t \boldsymbol{g}(\boldsymbol{y}^t, \boldsymbol{y}^{t-1}) + \sum_{t=2}^{T} (1 - \gamma^t)\boldsymbol{\mu}_t \tag{4.6}$$

in between the observed network statistics $\sum_{t=2}^{T} \boldsymbol{g}(\boldsymbol{y}^t, \boldsymbol{y}^{t-1})$ and the estimated network statistics $\sum_{t=2}^{T} \boldsymbol{\mu}_t$ from MCMC samples drawn from $P(\boldsymbol{y}^t|\boldsymbol{y}^{t-1}; \boldsymbol{\eta}_0)$. Instead of the difference between $\sum_{t=2}^{T} \boldsymbol{\mu}_t$ and $\sum_{t=2}^{T} \boldsymbol{g}(\boldsymbol{y}^t, \boldsymbol{y}^{t-1})$, we limit the amount of parameter update based on the difference between $\sum_{t=2}^{T} \boldsymbol{\mu}_t$ and $\hat{\boldsymbol{\xi}}(\boldsymbol{y})$ in each learning iteration. Empirically, the step length

γ^t is helpful in dampening the possibly drastic variations across different time intervals, for a stable parameter update when searching for an initial configuration.

Thus, we sequentially update the parameter in the direction of the MLE, while maintaining the approximated log-likelihood ratio $\hat{r}_s(\boldsymbol{\eta}, \boldsymbol{\eta}_0)$ estimated by MCMC samples is reasonably accurate. The closed-form solution for the maximizer of the approximated log-likelihood ratio at a specific learning iteration is

$$\tilde{\boldsymbol{\eta}} = \boldsymbol{\eta}_0 + \Big[\sum_{t=2}^{T} \boldsymbol{\Sigma}_t\Big]^{-1}\Big[\hat{\boldsymbol{\xi}}(\boldsymbol{y}) - \sum_{t=2}^{T} \boldsymbol{\mu}_t\Big].$$

Once an initial configuration is obtained from maximizing the approximated log-likelihood ratio, we proceed to the Newton-Raphson method to further update the parameter near its convergence. In this two-phase fusion where each component performs its designated task, both procedures require shorter learning iterations and undertake smaller computational burdens than only on their own.

4.4.3 MCMC for Dynamic Valued Networks

Fitting PST ERGM can be heavily dependent on MCMC sampling. In this section, we introduce the Metropolis-Hastings algorithm for drawing $\boldsymbol{y}^t$ conditional on $\boldsymbol{y}^{t-1}$. The superscript t is omitted for $\boldsymbol{y}^t$, $\boldsymbol{y}^{+,t}$, $\boldsymbol{y}^{-,t}$, and m^t to facilitate notational simplicity, as MCMC sampling is performed within a particular time point t. Additionally, we use a superscript k to refer to the current MCMC iteration.

In practice, valued networks are often sparse. To accommodate the sparsity of networks, as our proposal distribution, we employ a zero-inflated Poisson distribution that is also used in `ergm.count` [Kri19], an `R` library for static valued networks:

$$P(\boldsymbol{y}_{ij}^{k+1}; \lambda, \pi_0) = \begin{cases} \pi_0 + (1-\pi_0)\exp(-\lambda) & \text{if } \boldsymbol{y}_{ij}^{k+1} = 0; \\ (1-\pi_0)\exp(-\lambda) \times \lambda^{\boldsymbol{y}_{ij}^{k+1}}/\boldsymbol{y}_{ij}^{k+1}! & \text{if } \boldsymbol{y}_{ij}^{k+1} \in \mathbb{N}, \end{cases} \tag{4.7}$$

where $\lambda = \boldsymbol{y}_{ij}^{k} + 0.5$ and $\pi_0 \in [0, 1)$ is a pre-defined probability for the proposed dyad jumping

to 0. The 0.5 in λ prevents the proposed $\boldsymbol{y}_{ij}^{k+1}$ from locking into 0 when $\boldsymbol{y}_{ij}^{k} = 0$, and the proposed (i, j) is chosen randomly. We let $\pi_0 = 0.2$, a default value for the Poisson proposal distribution used in `ergm.count`, and it can be adjusted based on the user's prior knowledge on the sparsity of networks. The acceptance ratio α for the proposed $\boldsymbol{y}_{ij}^{k+1}$ is

$$q \times \frac{\boldsymbol{y}_{ij}^{+,k}!}{\boldsymbol{y}_{ij}^{+,k+1}!} \exp[\boldsymbol{\eta}^{+} \cdot \Delta \boldsymbol{g}^{+}(\boldsymbol{y}^{+}, \boldsymbol{y}^{t-1})_{ij}] \times \binom{m}{\boldsymbol{y}_{ij}^{-,k+1}} / \binom{m}{\boldsymbol{y}_{ij}^{-,k}} \exp[\boldsymbol{\eta}^{-} \cdot \Delta \boldsymbol{g}^{-}(\boldsymbol{y}^{-}, \boldsymbol{y}^{t-1})_{ij}], \quad (4.8)$$

where $\boldsymbol{y}^{+}$ and $\boldsymbol{y}^{-}$ are constructed from the observed $\boldsymbol{y}^{t-1}$ and the proposed network at MCMC iteration $k+1$. The change statistics $\Delta \boldsymbol{g}^{+}(\boldsymbol{y}^{+}, \boldsymbol{y}^{t-1})_{ij}$ denote the difference between $\boldsymbol{g}^{+}(\boldsymbol{y}^{+}, \boldsymbol{y}^{t-1})$ with $\boldsymbol{y}_{ij}^{+} = \boldsymbol{y}_{ij}^{+,k+1}$ and $\boldsymbol{g}^{+}(\boldsymbol{y}^{+}, \boldsymbol{y}^{t-1})$ with $\boldsymbol{y}_{ij}^{+} = \boldsymbol{y}_{ij}^{+,k}$ while rest of the $\boldsymbol{y}^{+}$ remains the same. The change statistics $\Delta \boldsymbol{g}^{-}(\boldsymbol{y}^{-}, \boldsymbol{y}^{t-1})_{ij}$ are calculated similarly except for notational difference, and the transition probability ratio q is

$$q = P(\boldsymbol{y}_{ij}^{k}; \lambda = \boldsymbol{y}_{ij}^{k+1} + 0.5, \pi_0) / P(\boldsymbol{y}_{ij}^{k+1}; \lambda = \boldsymbol{y}_{ij}^{k} + 0.5, \pi_0).$$

In this context, we propose a dyad value from the space of $\boldsymbol{y}^{t}$, but we decide to accept the proposed dyad value based on the construction of increment network $\boldsymbol{y}^{+,t}$ and decrement network $\boldsymbol{y}^{-,t}$, namely the dynamics between time $t-1$ and t. As we consolidate the temporal aspect into the PST ERGM, MCMC sampling becomes especially important in forecasting future networks besides its primary usage in parameter learning. Conditioning on the last observed network $\boldsymbol{y}^{T}$ under first order Markov assumption, we can forecast $\hat{\boldsymbol{y}}^{T+1}$ given the learned parameters $\tilde{\boldsymbol{\eta}}^{+}$ and $\tilde{\boldsymbol{\eta}}^{-}$ with the above scheme.

4.4.3.1 Contrastive Divergence Sampling

[Hum11] applied a K-step Contrastive Divergence (CD_K) sampling, an abridged MCMC, to speed up parameter estimation for static binary networks. Introduced in [Hin02] and [CH05], and applied to ERGM in [Fel14] and [Kri17b], the Contrastive Divergence (CD) for ERGM is formulated as

$$\mathrm{CD}_K = \mathrm{KL}[P_{\mathrm{data}}(\boldsymbol{y}^{\mathrm{obs}}) \parallel P_{\infty}(\boldsymbol{y})] - \mathrm{KL}[P_K(\boldsymbol{y}) \parallel P_{\infty}(\boldsymbol{y})].$$

The $P_{\text{data}}(\boldsymbol{y}^{\text{obs}})$ is the distribution of the observed data, $P_\infty(\boldsymbol{y})$ is the true model distribution, and $P_K(\boldsymbol{y})$ is the distribution of K-step MCMC samples [Hum11]. The gradient of CD_K for minimization given as

$$\nabla \text{CD}_K = \boldsymbol{g}(\boldsymbol{y}^{\text{obs}}) - \mathbb{E}_K[\boldsymbol{g}(\boldsymbol{y})] = \mathbf{0}$$

builds the foundation of CD_K sampling, where $\mathbb{E}_K[\boldsymbol{g}(\boldsymbol{y})]$ is the expected network statistics under the distribution of K-step MCMC samples.

In CD_K sampling, each sampled network is generated after K transitions starting from the observed network, so a burn-in phase is not required, and a tremendous sample size is not indispensable. Moreover, a small value of K can be used. In seeding an initial configuration via maximizing the approximated log-likelihood ratio, the CD_K sampling is in favor of the normality approximation, since each sampled network is at most K dyads different from the observed network. In the second phase where the learned parameter is close to the MLE, the network statistics of the MCMC samples are close to those of the observed networks. However, a small value of K generates a pseudo-observation of (4.6) that is not distinct from $\sum_{t=2}^{T} \boldsymbol{g}(\boldsymbol{y}^t, \boldsymbol{y}^{t-1})$, which may compromise the advantage of the partial stepping. Hence, a trade-off among the number of transitions, sample size, and learning iteration is needed. See [Kri17b] for a detailed study of using CD in ERGM fitting, especially regarding the choice of hyperparameters and stopping criterion.

4.5 Experiments

In this section, we apply PST ERGM to simulated and real data, for demonstrative purpose. Though the following real data can be analyzed by other frameworks such as Stochastic Actor-Oriented Model [Sni01] or Relational Event Model [But08b], we focus on the structure of dynamic valued networks, instead of the instantaneous action emitted by an actor given time-ordered sequences of historical events. In practice, when a network with relational strengths between nodes is observed at multiple time points, we can apply PST ERGM to

investigate the significance of network structures over time, especially those that can signify the network generating process.

The network statistics of interest are chosen from an extensive list in `ergm` [HHB22], an `R` library for network analysis. In this demonstration, the choice of network statistics in the increment process is identical to that in the decrement process. For real data experiments, the detail of the implementation and the formulations of selected network statistics are provided in 4.7.4.

4.5.1 Simulation Study

In this simulation with $T = 10$ and $n = 50$, we test our Metropolis-Hastings algorithm by generating $\boldsymbol{y}^2, \ldots, \boldsymbol{y}^T$ given $\boldsymbol{y}^1$ with pre-defined parameters $\boldsymbol{\eta}^+$ and $\boldsymbol{\eta}^-$. We then test our parameter learning algorithms by estimating the coefficients from the artificial data to compare with the true parameters. We choose four network statistics, (1) edge sum, (2) zeros, (3) mutuality, (4) transitive weight for both increment and decrement processes. The $\boldsymbol{g}^+(\boldsymbol{y}^{+,t}, \boldsymbol{y}^{t-1}) \in \mathbb{R}^4$ is specified as follows:

$$\Big(\sum_{ij} \boldsymbol{y}_{ij}^{+,t},\ \sum_{ij} \mathbb{1}(\boldsymbol{y}_{ij}^{+,t} = 0),\ \sum_{i<j} \sqrt{\boldsymbol{y}_{ij}^{+,t} \boldsymbol{y}_{ji}^{+,t}},\ \sum_{ij} \min(\boldsymbol{y}_{ij}^{+,t}, \max_{k \in N}(\min(\boldsymbol{y}_{ik}^{+,t}, \boldsymbol{y}_{kj}^{+,t}))) \Big).$$

The $\boldsymbol{g}^-(\boldsymbol{y}^{-,t}, \boldsymbol{y}^{t-1}) \in \mathbb{R}^4$ is specified similarly except for notational differences.

We initialize $\boldsymbol{\eta}^+ = (-2, 2, 1, 1)$, $\boldsymbol{\eta}^- = (-1, 2, 1, 1)$, and the maximum dyad value for decrement networks $m^t = 3$ for $t = 2, \ldots, T$. To ensure the networks are sampled with reasonable mixing and do not depend on initialization, each sampled $\boldsymbol{y}^t$ is generated after $20 \times n \times n$ MCMC transitions starting from an empty network. We repeat the process until we have 50 sequences of $\boldsymbol{y}^1, \cdots, \boldsymbol{y}^T$. As shown in Figure 4.4, the simulated network statistics have converged both within and across time points. In particular, the simulated networks are designed to be sparse, about 80% of the dyad values in $\boldsymbol{y}^t$ are zeros.

We then learn the parameter of PST ERGM for each generated sequence. To seed an initial configuration, we apply 20 iterations of partial stepping starting from a zero vector.

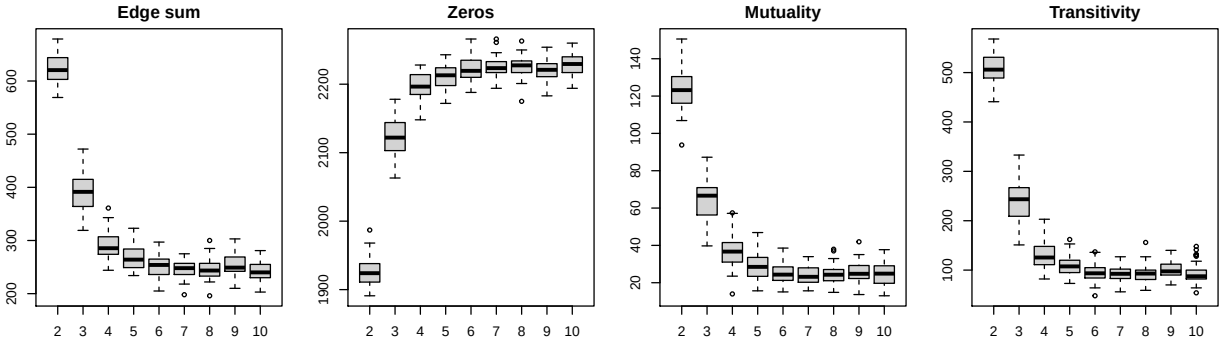

Figure 4.4: The distributions of network statistics based on 50 generated sequences of networks. The network statistics are evaluated at $\boldsymbol{y}^t$ from $t = 2$ to 10.

An MCMC sample size of 100 with CD_n sampling is used for each time t. The number of MCMC transitions is set to n. Subsequently, to refine the parameter, we apply 5 iterations of Newton-Raphson method, where an MCMC sample size of 1000 with CD_n sampling is used for each time t. The medians and standard deviations of $|\tilde{\boldsymbol{\eta}}^+ - \boldsymbol{\eta}^+|$ over 50 estimations are reported in Table 4.1. The corresponding results for $|\tilde{\boldsymbol{\eta}}^- - \boldsymbol{\eta}^-|$ are reported in Table 4.2.

# of time step & node	$\lvert\tilde{\boldsymbol{\eta}}_1^+ - \boldsymbol{\eta}_1^+\rvert$	$\lvert\tilde{\boldsymbol{\eta}}_2^+ - \boldsymbol{\eta}_2^+\rvert$	$\lvert\tilde{\boldsymbol{\eta}}_3^+ - \boldsymbol{\eta}_3^+\rvert$	$\lvert\tilde{\boldsymbol{\eta}}_4^+ - \boldsymbol{\eta}_4^+\rvert$
$T = 10, n = 50$	0.0027 (0.065)	0.0046 (0.087)	0.0021 (0.056)	0.0128 (0.064)

Table 4.1: The medians (standard deviation) of $|\tilde{\boldsymbol{\eta}}^+ - \boldsymbol{\eta}^+|$ over 50 estimations.

# of time step & node	$\lvert\tilde{\boldsymbol{\eta}}_1^- - \boldsymbol{\eta}_1^-\rvert$	$\lvert\tilde{\boldsymbol{\eta}}_2^- - \boldsymbol{\eta}_2^-\rvert$	$\lvert\tilde{\boldsymbol{\eta}}_3^- - \boldsymbol{\eta}_3^-\rvert$	$\lvert\tilde{\boldsymbol{\eta}}_4^- - \boldsymbol{\eta}_4^-\rvert$
$T = 10, n = 50$	0.0471 (0.169)	0.0228 (0.182)	0.0108 (0.162)	0.0033 (0.076)

Table 4.2: The medians (standard deviation) of $|\tilde{\boldsymbol{\eta}}^- - \boldsymbol{\eta}^-|$ over 50 estimations.

On average, the estimations are close to the true parameters as the medians of absolute differences are close to zeros. We also check if the 95% confidence intervals of the learned parameters cover the true parameters. Figure 4.5 displays the confidence intervals $\tilde{\boldsymbol{\eta}}_i \pm 1.96\tilde{\boldsymbol{\sigma}}_i$ for the 50 estimations, where $\tilde{\boldsymbol{\sigma}}_i$ denotes the standard error of $\tilde{\boldsymbol{\eta}}_i$. The standard errors are

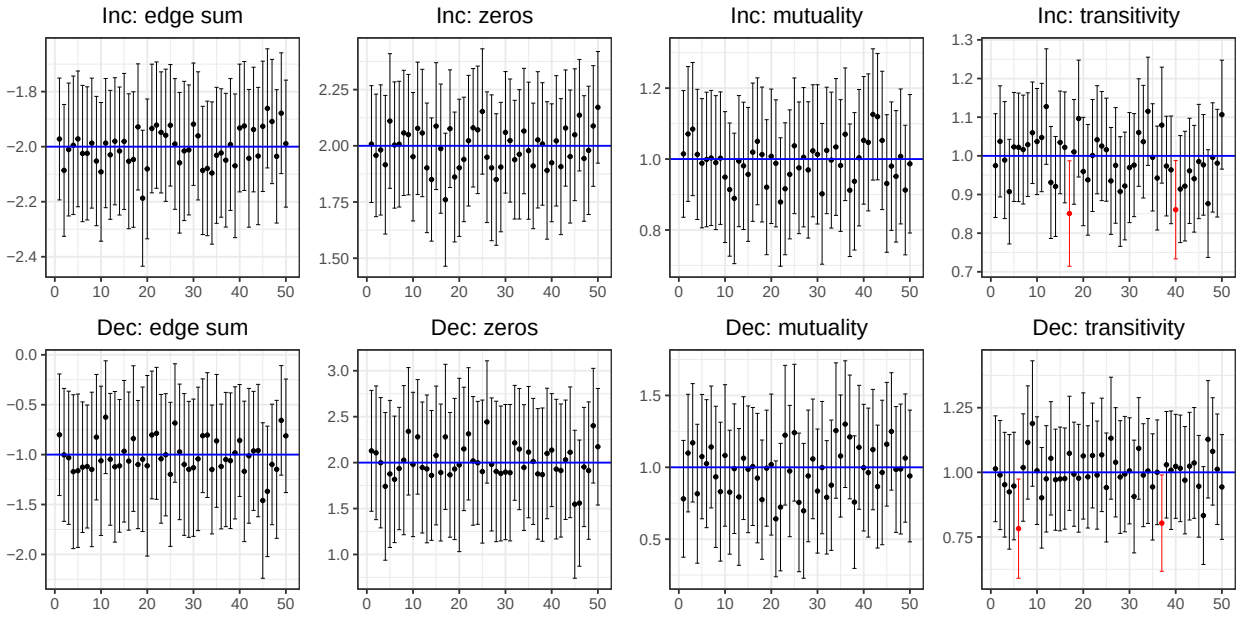

Figure 4.5: The 95% confidence intervals (black bars) of the 50 learned parameters (dots) for each network statistic. The blue horizontal lines indicate the true parameter values $\boldsymbol{\eta}^+ = (-2, 2, 1, 1)$ and $\boldsymbol{\eta}^- = (-1, 2, 1, 1)$. The red bars indicate the confidence intervals that do not cover the true parameters.

obtained from the Fisher Information matrix $\boldsymbol{I}(\boldsymbol{\eta}) \approx -\hat{\boldsymbol{H}}(\tilde{\boldsymbol{\eta}})$ of (4.5) evaluated at the learned parameter $\tilde{\boldsymbol{\eta}}$ with 100 sampled networks for each t. Each sampled network is generated after 1000 MCMC transitions starting from the observed $\boldsymbol{y}^t$. We notice that the true parameters are covered by the confidence intervals most of the time.

4.5.2 Modeling: Students Contact Networks

[MFB15] used wearable sensors to detect face-to-face contacts between students among nine classes in a high school. The real-time contact events were logged for every 20-second interval of any two students within a physical distance of 1.5 meters from 02-Dec-2013 to 06-Dec-2013. Additionally, online social network (Facebook) was submitted by the students voluntarily. In this demonstration, we model student interactions within one of the nine classes, whose

class name is MP. There are $n = 29$ students which consist of 11 females and 18 males. We divide the entries by day to construct $T = 5$ undirected valued networks, where $\boldsymbol{y}_{ij}^t$ is the number of unique contacts between student i and student j on day t. The duration of each contact can be different and expansive. The nodal covariate $\boldsymbol{x}_i \in \{\mathrm{F}, \mathrm{M}\}$ is the gender of student i, and dyadic covariate $\boldsymbol{z}_{ij} \in \{1, 0\}$ indicates whether student i and student j are friends on Facebook or not.

We choose six network statistics of interest for analysis, and we learn a time-heterogeneous PST ERGM $\prod_{t=2}^{5} P(\boldsymbol{y}^t|\boldsymbol{y}^{t-1}; \boldsymbol{\eta}^t)$ for the data. A time-homogeneous model was attempted, but the large variation between different intervals suggests that a time-heterogeneous model is appropriate and realistic. The estimated coefficients and standard errors for the increment process are reported in Table 4.3. The corresponding results for the decrement process are reported in Table 4.4.

Network Statistics	$\boldsymbol{\eta}^{+,2}$	$\boldsymbol{\eta}^{+,3}$	$\boldsymbol{\eta}^{+,4}$	$\boldsymbol{\eta}^{+,5}$
Edge sum	**3.215** (0.092)	**3.316** (0.097)	**3.023** (0.111)	**3.197** (0.079)
Dispersion	**-6.825** (0.231)	**-7.576** (0.201)	**-6.470** (0.222)	**-6.506** (0.201)
Homophily (M)	**0.151** (0.073)	**0.223** (0.079)	0.096 (0.086)	**0.135** (0.059)
Heterophily (M-F)	0.042 (0.068)	0.083 (0.080)	0.067 (0.082)	0.056 (0.057)
Facebook	-0.005 (0.042)	**0.123** (0.031)	-0.033 (0.047)	**0.092** (0.035)
Transitive weight	**-0.142** (0.038)	**-0.068** (0.026)	-0.015 (0.044)	**-0.187** (0.032)

Coefficients statistically significant at 0.05 level are bolded.

Table 4.3: The parameter estimation (standard error) of $\boldsymbol{\eta}^{+,t}$ for the students contact networks.

The positive coefficients on the edge sum term in the increment process from $t = 2$ to 5 indicate frequent interactions among students throughout the week. However, the negative coefficients on the edge sum term in the decrement process from $t = 3$ to 5 suggest a short duration on the increase of contact occurrences. In other words, the number of contacts

Network Statistics	$\boldsymbol{\eta}^{-,2}$	$\boldsymbol{\eta}^{-,3}$	$\boldsymbol{\eta}^{-,4}$	$\boldsymbol{\eta}^{-,5}$
Edge sum	**0.432** (0.214)	**-0.980** (0.220)	**-0.630** (0.183)	-0.294 (0.162)
Dispersion	**-6.572** (0.277)	**-6.144** (0.323)	**-6.102** (0.275)	**-6.680** (0.270)
Homophily (M)	-0.161 (0.184)	**1.344** (0.213)	**0.665** (0.189)	**0.390** (0.150)
Heterophily (M-F)	-0.222 (0.190)	**0.481** (0.174)	-0.205 (0.189)	0.100 (0.152)
Facebook	-0.083 (0.088)	**0.231** (0.105)	**0.193** (0.088)	0.099 (0.079)
Transitive weight	0.087 (0.066)	**-0.445** (0.075)	**-0.446** (0.044)	-0.077 (0.045)

Coefficients statistically significant at 0.05 level are bolded.

Table 4.4: The parameter estimation (standard error) of $\boldsymbol{\eta}^{-,t}$ for the students contact networks.

fluctuates over time, which supports the adoption of a time-heterogeneous model. Similarly, the highly negative coefficients on the dispersion term in both increment and decrement processes suggest a strong degree of over-dispersion in the number of contacts. This can be verified by the standard deviation of non-zero dyad values across five days, which is 10.2, whereas the mean is about half in magnitude, which is 5.8.

In the increment process from $t = 2$ to 5, the positive coefficients on the homophily for males suggest a strong gender effect in promoting more interactions among male students. Additionally, the positive coefficients in the decrement process from $t = 3$ to 5 indicate that the active interactions among males tend to be ongoing once they have begun. However, these effects are less significant for students of different genders, given the imbalanced proportion between females and males. As supporting evidence, about 89% or 136 out of $\binom{18}{2} = 153$ pairs of male students have contact events logged by sensors, and their average span is 2.8 days. In contrast, about 71% or 141 out of 198 pairs of male and female students have contact events logged, and their average span is 2.3 days.

Furthermore, in the increment process from $t = 2$ to 5, the alternating signs of coefficients on the Facebook term indicate that if two students are friends online, they occasionally have

active interactions in school. However, the majority of positive coefficients on the Facebook term in the decrement process suggest that online friendships maintain offline interactions. About 82% or 119 out of 145 reported Facebook friendships have contact events logged in school, and their average span is 2.9 days. Lastly, the transitive relationship in the number of contacts is weak, as indicated by the negative coefficients in the increment process. The majority of negative coefficients in the decrement process suggests that the transitivity tends not to persist over time.

To validate the learned model heuristically, we simulate networks with the estimated parameters to compare the sampled network statistics with the observed network statistics. For $t = 2, \ldots, 5$, we generate 100 valued networks $\boldsymbol{y}^t$ conditional on the observed $\boldsymbol{y}^{t-1}$, where each sampled network is generated after $200 \times n \times n$ MCMC transitions starting from an empty network. We then construct the corresponding $\boldsymbol{y}^{+,t}$ and $\boldsymbol{y}^{-,t}$ and calculate their network statistics. The distributions of the simulated network statistics and the observed network statistics values are displayed in Figure 4.6. Overall, the simulated network statistics align with the observed values, suggesting the learned PST ERGM is a good representation of the observed data in terms of the six selected network statistics.

4.5.3 Forecasting: Baboons Interaction Networks

[GGP20] studied the interactions among $n = 13$ baboons for a duration of 28 days. The contact events are recorded by sensors for every 20-second interval of any two primates within proximity of 1.5 meters. The data is divided by day to construct a sequence of $T = 28$ undirected valued networks, where $\boldsymbol{y}_{ij}^t$ is the number of unique contacts between baboon i and baboon j on day t. The duration of each contact can be different and expansive. In this experiment, we learn a time-homogeneous PST ERGM based on the data from day 1 to 23, and we forecast 5 subsequent networks to compare with the observed network from day 24 to 28. Though a time-heterogeneous model can learn the transitions well, it lacks the ability to forecast future networks as a learned parameter $\boldsymbol{\eta}^t$ is tailored to the designated

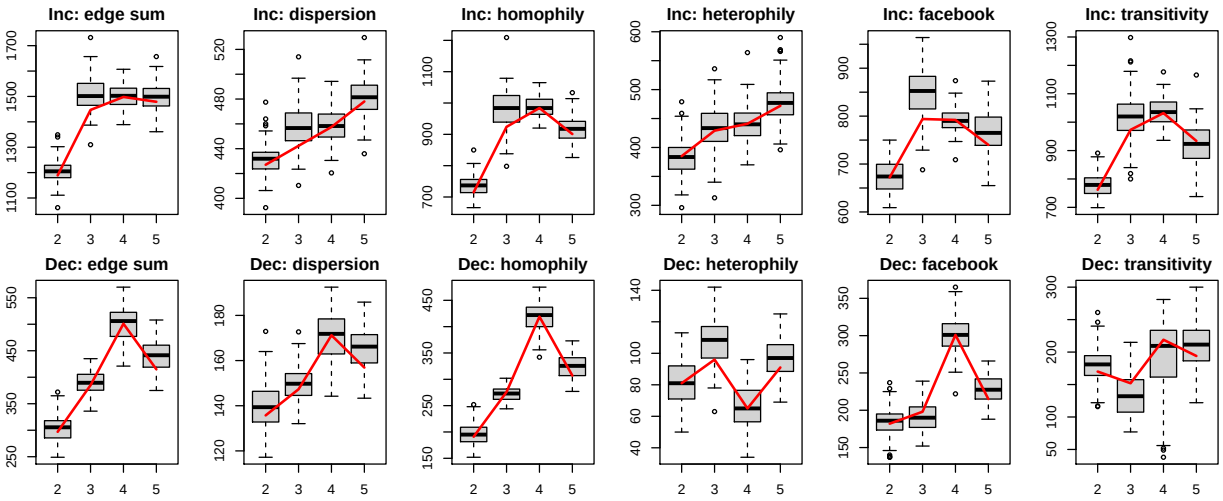

Figure 4.6: The distribution of the sampled network statistics (box plots) and the observed network statistics values (red lines) across four consecutive intervals for increment (Inc) and decrement (Dec) processes.

time point t and cannot be extended to the next time point $t+1$.

We choose four network statistics for this task. The estimated parameters and standard errors for both increment and decrement processes are reported in Table 4.5. To forecast out-of-sample data, we generate 100 valued network $\hat{\boldsymbol{y}}^t$ conditional on the observed $\boldsymbol{y}^{t-1}$ for $t = 24, \cdots, 28$. Each sampled network is generated after $200 \times n \times n$ MCMC transitions starting from an empty network. We then construct the corresponding $\hat{\boldsymbol{y}}^{+,t}$ and $\hat{\boldsymbol{y}}^{-,t}$ and calculate their network statistics. The distributions of the forecasted network statistics and the observed network statistics values are displayed in Figure 4.7.

In exchange for the extrapolation of future temporal trends, the time-homogeneous PST ERGM that consolidates the fluctuation throughout 23 days into one parameter $\boldsymbol{\eta}$ may introduce variation to the forecasted network statistics. The discrepancy on day 25 in Figure 4.7 is potentially impacted by this outcome. Furthermore, the discrepancy of the propensity term on day 27 in the decrement process may be influenced by the increase of all network statistics from day 26, as our proposed PST ERGM allows interaction between the two pro-

Network Statistics	$\boldsymbol{\eta}^+$	$\boldsymbol{\eta}^-$
Edge sum	**4.674** (0.017)	**-0.160** (0.014)
Propensity	**9.937** (0.204)	**10.345** (0.154)
Dispersion	**-14.728** (0.139)	**-14.064** (0.103)
Transitive weight	**-0.060** (0.007)	**-0.145** (0.007)

Coefficients statistically significant at 0.05 level are bolded.

Table 4.5: The parameter estimation (standard error) for the baboons interaction networks from day 1 to 23.

cesses. Note that the $\boldsymbol{y}^{+,t}$ and $\boldsymbol{y}^{-,t}$ in PST ERGM are no longer conditionally independent as the sample space of valued networks is infinite. In summary, besides prediction error for the unseen data, the learned time-homogeneous PST ERGM effectively recovers the sudden change on day 26 along with the temporal trends from day 24 to 28.

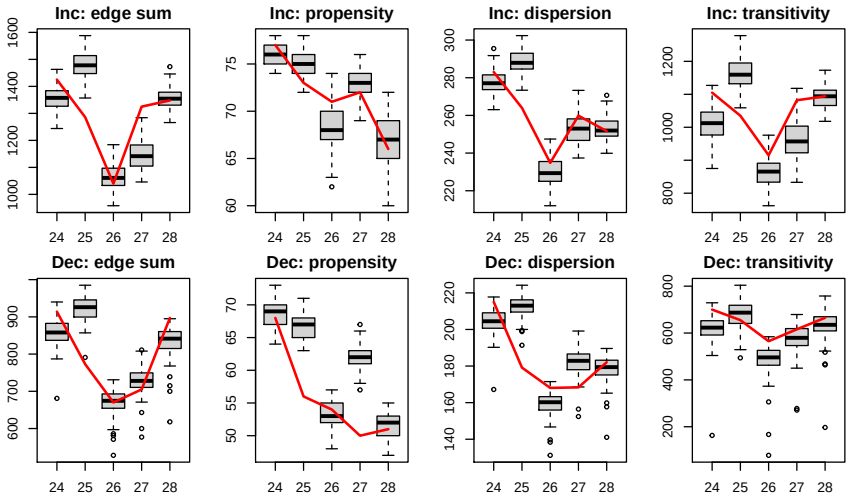

Figure 4.7: The distribution of the forecasted network statistics (box plots) and the observed network statistics values (red lines) for increment (Inc) and decrement (Dec) processes from day 24 to 28.

Another aspect worth mentioning is the comparison between the edge sum term and the

propensity term in this experiment. The propensity term $\sum_{i<j} \mathbb{1}(\boldsymbol{y}_{ij} > 0)$ is essentially a thresholding version of the edge sum term $\sum_{i<j} \boldsymbol{y}_{ij}$. We can dichotomize a valued network into a binary network and count the number of edges to calculate the propensity term. In the first column of Figure 4.7, the observed edge sums in red lines present a decreasing trend followed by an increasing trend in both processes from day 24 to 28. The dispersion term and the transitivity term that are also evaluated with the valued networks produce similar patterns. However, in the second column of Figure 4.7, the observed propensity terms in red lines primarily show a decreasing trend in both processes. Empirically, dichotomizing dynamic valued networks into dynamic binary networks, or dyad value thresholding, for network analysis may introduce biases [TB11] that result in unrealistic network dynamics.

4.6 Discussion

This chapter introduces a probabilistic model for dynamic valued networks. In practice, the factors and processes that increase relational strength are usually different from those that decrease relational strength. While dynamic network models should capture the intrinsic difference between consecutive networks, models neglecting the confounding effect of structural change may result in misinterpretation of network evolution. Inspired by [KH14], we propose a PST ERGM to dissect valued network transitions with two sets of intermediate networks, where one manages dyad value increment, and the other manages dyad value decrement. Our proposed PST ERGM provides the interpretability of network evolution and the capability to forecast temporal trends.

Several improvements to the PST ERGM are possible for future development. We can extend the sample space to networks with continuous dyad values. In this context, novel reference functions and network statistics are needed as PST ERGM becomes a continuous probability distribution. Furthermore, besides dyad value increment and decrement, alternative ways to dissect network evolution are permitted, as long as the confounding effect of network

dynamics is avoided. Over time, the number of participants and the process that induces their relations may not be fixed or completely observed. It is of great importance for a dynamic network model to identify the temporal changes punctually [PYP19, YMW21, KLC23] and to adjust the structural changes accordingly [KHM11].

Finally, model degeneracy which is studied theoretically by [HRS03] is a well-known challenge in the ERGM framework. In modeling dynamic valued networks, though an infinite sample space does not have a maximal graph on which a PST ERGM will concentrate, the MLE can be difficult to find by the MCMC methods. Also, the geometrically weighted statistics that are used to alleviate the degeneracy problem in fitting static binary networks are currently not available for valued networks. Therefore, a rigorous way to design more informative network statistics as in [SPR06], and a systematic way to evaluate the goodness of model fit as in [HGH08] are needed for dynamic valued networks. The tapered ERGM introduced in [FH17], and [BH22] can also be extended to the PST ERGM to alleviate the degeneracy issue.

4.7 Appendix

4.7.1 Comparison of Simple Fitted Models

In this section, we present the simple fitted models that motivate the increment and decrement networks in Section 4.3.1. First, we fit a Poisson-reference valued ERGM [Kri12a] that involves only the edge sum term $\boldsymbol{g}(\boldsymbol{y}^t) = \sum_{(i,j)\in\mathbb{Y}} \boldsymbol{y}_{ij}^t$ to $\boldsymbol{y}^{25}$ and $\boldsymbol{y}^{26}$, respectively. The coefficients and standard errors are displayed in Table 4.6. We notice the two coefficients are similar, providing little information about the transition. Furthermore, we fit a PST ERGM that involves only the edge sum terms $\boldsymbol{g}(\boldsymbol{y}^t, \boldsymbol{y}^{t-1}) = \big(\sum_{(i,j)\in\mathbb{Y}} \boldsymbol{y}_{ij}^{+,t}, \sum_{(i,j)\in\mathbb{Y}} \boldsymbol{y}_{ij}^{-,t}\big)$ to both $\boldsymbol{y}^{+,26}$ and $\boldsymbol{y}^{-,26}$. The coefficients and standard errors are displayed in Table 4.7. The positive and negative coefficients reveal the fluctuation in dyad values between the observed time points.

Network Statistics	$\boldsymbol{\eta}$ for $\boldsymbol{g}(\boldsymbol{y}^{25})$	$\boldsymbol{\eta}$ for $\boldsymbol{g}(\boldsymbol{y}^{26})$
Edge sum	**2.371** (0.033)	**2.421** (0.035)

Table 4.6: The parameter estimation (standard error) for the baboons interaction networks on day 25 and day 26, respectively. Coefficients statistically significant at 0.05 level are bolded.

Network Statistics	$\boldsymbol{\eta}^{+}$	$\boldsymbol{\eta}^{-}$
Edge sum	**1.887** (0.055)	**-0.843** (0.069)

Table 4.7: The parameter estimation (standard error) for the baboons interaction networks, using the increment and decrement networks. Coefficients statistically significant at 0.05 level are bolded.

4.7.2 Special Cases of PST ERGM

In this section, we derive two special cases of PST ERGM, under a specific set of sufficient statistics and temporal information. Consider a simple PST ERGM with the edge sums of increment and decrement networks as two network statistics:

$$\boldsymbol{g}(\boldsymbol{y}^{t}, \boldsymbol{y}^{t-1}) = \big(\boldsymbol{g}^{+}(\boldsymbol{y}^{+,t}, \boldsymbol{y}^{t-1}), \boldsymbol{g}^{-}(\boldsymbol{y}^{-,t}, \boldsymbol{y}^{t-1})\big) = \Big(\sum_{(i,j)\in\mathbb{Y}} \boldsymbol{y}_{ij}^{+,t}, \sum_{(i,j)\in\mathbb{Y}} \boldsymbol{y}_{ij}^{-,t} \Big) \in \mathbb{R}^2.$$

Let $\mathcal{Y}^t(\boldsymbol{y}^{t-1}) \subseteq \{\boldsymbol{y}^t \in \mathbb{N}_0^{\mathbb{Y}} : \boldsymbol{y}_{ij}^t > \boldsymbol{y}_{ij}^{t-1} \ \forall \ (i,j) \in \mathbb{Y}\}$ be a sample space for $\boldsymbol{y}^t$ starting from $\boldsymbol{y}^{t-1}$. The increment network $\boldsymbol{y}^{+,t}$ is essentially the $\boldsymbol{y}^t$, and we have

$$\begin{aligned}
P(\boldsymbol{y}^t|\boldsymbol{y}^{t-1};\boldsymbol{\eta}) &= \frac{\big[\prod_{ij}(\boldsymbol{y}_{ij}^t!)^{-1}\exp(\boldsymbol{\eta}^+\cdot\boldsymbol{y}_{ij}^t)\big]\times\big[\prod_{ij}\binom{m^t}{\boldsymbol{y}_{ij}^{t-1}}\exp(\boldsymbol{\eta}^-\cdot\boldsymbol{y}_{ij}^{t-1})\big]}{\sum_{\boldsymbol{y}^t\in\mathcal{Y}^t(\boldsymbol{y}^{t-1})}\big[\prod_{ij}(\boldsymbol{y}_{ij}^t!)^{-1}\exp(\boldsymbol{\eta}^+\cdot\boldsymbol{y}_{ij}^t)\times\prod_{ij}\binom{m^t}{\boldsymbol{y}_{ij}^{t-1}}\exp(\boldsymbol{\eta}^-\cdot\boldsymbol{y}_{ij}^{t-1})\big]} \\
&= \frac{\prod_{ij}\binom{m^t}{\boldsymbol{y}_{ij}^{t-1}}\exp(\boldsymbol{\eta}^-\cdot\boldsymbol{y}_{ij}^{t-1})}{\prod_{ij}\binom{m^t}{\boldsymbol{y}_{ij}^{t-1}}\exp(\boldsymbol{\eta}^-\cdot\boldsymbol{y}_{ij}^{t-1})}\times\frac{\prod_{ij}(\boldsymbol{y}_{ij}^t!)^{-1}\exp(\boldsymbol{\eta}^+\cdot\boldsymbol{y}_{ij}^t)}{\sum_{\boldsymbol{y}^t\in\mathcal{Y}^t(\boldsymbol{y}^{t-1})}\big[\prod_{ij}(\boldsymbol{y}_{ij}^t!)^{-1}\exp(\boldsymbol{\eta}^+\cdot\boldsymbol{y}_{ij}^t)\big]} \\
&= \prod_{ij}\frac{(\boldsymbol{y}_{ij}^t!)^{-1}\exp(\boldsymbol{\eta}^+\cdot\boldsymbol{y}_{ij}^t)}{\sum_{u=\boldsymbol{y}_{ij}^{t-1}+1}^{\infty}\big[(u)^{-1}\exp(\boldsymbol{\eta}^+\cdot u)\big]} \\
&= \prod_{ij}\frac{(\boldsymbol{y}_{ij}^t!)^{-1}\exp(\boldsymbol{\eta}^+\cdot\boldsymbol{y}_{ij}^t)}{\exp\big(\exp(\boldsymbol{\eta}^+)\big)-\sum_{u=0}^{\boldsymbol{y}_{ij}^{t-1}}(u!)^{-1}\exp(\boldsymbol{\eta}^+\cdot u)} \\
&= \prod_{ij}\frac{P_{\text{Pois}}\big(\boldsymbol{y}_{ij}^t\big)}{1-\sum_{u=0}^{\boldsymbol{y}_{ij}^{t-1}}P_{\text{Pois}}(u)},
\end{aligned}$$

where $P_{\text{Pois}}(x)$ denotes the probability mass function of $\text{Poisson}\big(\lambda=\exp(\boldsymbol{\eta}^+)\big)$ evaluated at x. The PST ERGM $P(\boldsymbol{y}^t|\boldsymbol{y}^{t-1};\boldsymbol{\eta})$ for $\boldsymbol{y}^t \in \mathcal{Y}^t(\boldsymbol{y}^{t-1})$ becomes a dyadic independent truncated Poisson distribution.

Moreover, let $\mathcal{Y}^t(\boldsymbol{y}^{t-1}) \subseteq \{\boldsymbol{y}^t \in \mathbb{N}_0^{\mathbb{Y}} : \boldsymbol{y}_{ij}^t < \boldsymbol{y}_{ij}^{t-1} \leq m^t \ \forall \ (i,j) \in \mathbb{Y}\}$ be another sample space for $\boldsymbol{y}^t$. The decrement network $\boldsymbol{y}^{-,t}$ is essentially the $\boldsymbol{y}^t$, and we have

$$\begin{aligned}
P(\boldsymbol{y}^t|\boldsymbol{y}^{t-1};\boldsymbol{\eta}) &= \frac{\big[\prod_{ij}(\boldsymbol{y}_{ij}^{t-1}!)^{-1}\exp(\boldsymbol{\eta}^+\cdot\boldsymbol{y}_{ij}^{t-1})\big]\times\big[\prod_{ij}\binom{m^t}{\boldsymbol{y}_{ij}^t}\exp(\boldsymbol{\eta}^-\cdot\boldsymbol{y}_{ij}^t)\big]}{\sum_{\boldsymbol{y}^t\in\mathcal{Y}^t(\boldsymbol{y}^{t-1})}\big[\prod_{ij}(\boldsymbol{y}_{ij}^{t-1}!)^{-1}\exp(\boldsymbol{\eta}^+\cdot\boldsymbol{y}_{ij}^{t-1})\times\prod_{ij}\binom{m^t}{\boldsymbol{y}_{ij}^t}\exp(\boldsymbol{\eta}^-\cdot\boldsymbol{y}_{ij}^t)\big]} \\
&= \frac{\prod_{ij}(\boldsymbol{y}_{ij}^{t-1}!)^{-1}\exp(\boldsymbol{\eta}^+\cdot\boldsymbol{y}_{ij}^{t-1})}{\prod_{ij}(\boldsymbol{y}_{ij}^{t-1}!)^{-1}\exp(\boldsymbol{\eta}^+\cdot\boldsymbol{y}_{ij}^{t-1})}\times\frac{\prod_{ij}\binom{m^t}{\boldsymbol{y}_{ij}^t}\exp(\boldsymbol{\eta}^-\cdot\boldsymbol{y}_{ij}^t)}{\sum_{\boldsymbol{y}^t\in\mathcal{Y}^t(\boldsymbol{y}^{t-1})}\big[\prod_{ij}\binom{m^t}{\boldsymbol{y}_{ij}^t}\exp(\boldsymbol{\eta}^-\cdot\boldsymbol{y}_{ij}^t)\big]} \\
&= \prod_{ij}\frac{\binom{m^t}{\boldsymbol{y}_{ij}^t}\exp(\boldsymbol{\eta}^-\cdot\boldsymbol{y}_{ij}^t)}{\sum_{u=0}^{\boldsymbol{y}_{ij}^{t-1}-1}\big[\binom{m^t}{u}\exp(\boldsymbol{\eta}^-\cdot u)\big]} \\
&= \prod_{ij}\frac{\binom{m^t}{\boldsymbol{y}_{ij}^t}\exp(\boldsymbol{\eta}^-\cdot\boldsymbol{y}_{ij}^t)}{\big(1+\exp(\boldsymbol{\eta}^-)\big)^{m^t}-\sum_{u=\boldsymbol{y}_{ij}^{t-1}}^{m^t}\binom{m^t}{u}\exp(\boldsymbol{\eta}^-\cdot u)} \\
&= \prod_{ij}\frac{P_{\text{Bino}}\big(\boldsymbol{y}_{ij}^t\big)}{1-\sum_{u=\boldsymbol{y}_{ij}^{t-1}}^{m^t}P_{\text{Bino}}\big(u\big)},
\end{aligned}$$

where $P_{\text{Bino}}(x)$ denotes the probability mass function of Binomial$\big(m^t, p = \text{logit}^{-1}(\boldsymbol{\eta}^-)\big)$ evaluated at x. The PST ERGM $P(\boldsymbol{y}^t|\boldsymbol{y}^{t-1};\boldsymbol{\eta})$ for $\boldsymbol{y}^t \in \mathcal{Y}^t(\boldsymbol{y}^{t-1})$ becomes a dyadic independent truncated Binomial distribution.

4.7.3 Parameter Estimation Algorithms

The partial stepping algorithm of PST ERGM parameter estimation to seed an initial configuration is provided in Algorithm 3. In this work, we let γ_c^t be the ratio of the current iteration c to the maximum number of iterations C for each time t. Only in the last learning iteration where $\gamma_c^t = 1$ do we use the difference between observed network statistics $\sum_{t=2}^T \boldsymbol{g}(\boldsymbol{y}^t, \boldsymbol{y}^{t-1})$ and estimated network statistics $\sum_{t=2}^T \boldsymbol{\mu}_t$ to update the parameter.

Algorithm 3 Partial stepping algorithm

Input: initialized parameter $\boldsymbol{\eta}_0$, learning iteration C, sample size s, $\{\boldsymbol{y}^1, \cdots, \boldsymbol{y}^T\}$
for $c = 1, \cdots, C$ **do**
  for $t = 2, \cdots, T$ **do**
    Generate s MCMC samples $\boldsymbol{y}_1^t, \cdots, \boldsymbol{y}_s^t$ from $P(\boldsymbol{y}^t|\boldsymbol{y}^{t-1};\boldsymbol{\eta}_{c-1})$ as in Section 4.4.3
    Calculate $\boldsymbol{\mu}_t = s^{-1}\sum_{i'=1}^s \boldsymbol{g}(\boldsymbol{y}_{i'}^t, \boldsymbol{y}^{t-1})$
    Calculate $\boldsymbol{\Sigma}_t = s^{-1}\sum_{i'=1}^s \boldsymbol{g}(\boldsymbol{y}_{i'}^t, \boldsymbol{y}^{t-1})\boldsymbol{g}(\boldsymbol{y}_{i'}^t, \boldsymbol{y}^{t-1})^\top - \boldsymbol{\mu}_t\boldsymbol{\mu}_t^\top$
  end for
  For $\gamma_c^t = c/C$, calculate $\hat{\boldsymbol{\xi}}(\boldsymbol{y}) = \sum_{t=2}^T \gamma_c^t\, \boldsymbol{g}(\boldsymbol{y}^t, \boldsymbol{y}^{t-1}) + \sum_{t=2}^T (1-\gamma_c^t)\boldsymbol{\mu}_t$
  $\boldsymbol{\eta}_c = \boldsymbol{\eta}_{c-1} + [\sum_{t=2}^T \boldsymbol{\Sigma}_t]^{-1}[\hat{\boldsymbol{\xi}}(\boldsymbol{y}) - \sum_{t=2}^T \boldsymbol{\mu}_t]$
end for
$\tilde{\boldsymbol{\eta}} \leftarrow \boldsymbol{\eta}_c$
Output: learned parameter $\tilde{\boldsymbol{\eta}}$

Once a parameter is obtained from maximizing the approximated log-likelihood ratio, we proceed to the Newton-Raphson method to further update the parameter near its convergence. The Newton-Raphson algorithm of PST ERGM parameter estimation is provided

in Algorithm 4. The CD_K sampling algorithm to generate a single sampled network $\boldsymbol{y}_{i'}^t$ is provided in Algorithm 5, and it is used in Step 4 of Algorithm 3 and Step 4 of Algorithm 4.

Algorithm 4 Newton-Raphson algorithm

1: **Input**: initialized parameter $\boldsymbol{\eta}_0$, learning iteration C, sample size s, $\{\boldsymbol{y}^1, \cdots, \boldsymbol{y}^T\}$
2: **for** $c = 1, \cdots, C$ **do**
3: **for** $t = 2, \cdots, T$ **do**
4: Generate s MCMC samples $\boldsymbol{y}_1^t, \cdots, \boldsymbol{y}_s^t$ from $P(\boldsymbol{y}^t | \boldsymbol{y}^{t-1}; \boldsymbol{\eta}_{c-1})$ as in Section 4.4.3
5: Calculate $\boldsymbol{\mu}_t = s^{-1} \sum_{i'=1}^{s} \boldsymbol{g}(\boldsymbol{y}_{i'}^t, \boldsymbol{y}^{t-1})$
6: Calculate $\boldsymbol{\Sigma}_t = s^{-1} \sum_{i'=1}^{s} \boldsymbol{g}(\boldsymbol{y}_{i'}^t, \boldsymbol{y}^{t-1}) \boldsymbol{g}(\boldsymbol{y}_{i'}^t, \boldsymbol{y}^{t-1})^\top - \boldsymbol{\mu}_t \boldsymbol{\mu}_t^\top$
7: **end for**
8: Calculate $\hat{\boldsymbol{S}}(\boldsymbol{\eta}_{c-1}) = \sum_{t=2}^{T} [\boldsymbol{g}(\boldsymbol{y}^t, \boldsymbol{y}^{t-1}) - \boldsymbol{\mu}_t]$ and $\hat{\boldsymbol{H}}(\boldsymbol{\eta}_{c-1}) = -\sum_{t=2}^{T} \boldsymbol{\Sigma}_t$
9: $\boldsymbol{\eta}_c = \boldsymbol{\eta}_{c-1} - \hat{\boldsymbol{H}}(\boldsymbol{\eta}_{c-1})^{-1} \hat{\boldsymbol{S}}(\boldsymbol{\eta}_{c-1})$
10: **end for**
11: $\tilde{\boldsymbol{\eta}} \leftarrow \boldsymbol{\eta}_c$
12: **Output**: learned parameter $\tilde{\boldsymbol{\eta}}$

The complexity of Algorithm 5 to sample one valued network with K MCMC transitions is $O(K\mathcal{C}_\alpha)$. The term $\mathcal{C}_\alpha$ is the complexity of calculating the acceptance ratio α with Equation (4.8), which depends on the choice of network statistics for both increment and decrement processes as well as the randomness of the proposed dyad values from Equation (4.7). Likewise, for Algorithm 3, the complexity to sample s valued networks in Step 4 is $O(sK\mathcal{C}_\alpha)$. In Steps 5 and 6 of Algorithm 3, the complexity of calculating the mean vector $\boldsymbol{\mu}_t \in \mathbb{R}^p$ and the covariance matrix $\boldsymbol{\Sigma}_t \in \mathbb{R}^{p \times p}$ of s sampled network statistics is $O(s\mathcal{C}_{2g} + sp + sp^2)$. The term $\mathcal{C}_{2g}$ is the complexity to calculate the two network statistics $\boldsymbol{g}(\boldsymbol{y}^t, \boldsymbol{y}^{t-1})$ that are based on the user's choice. Finally, the complexity of calculating the pseudo-observation $\hat{\boldsymbol{\xi}}(\boldsymbol{y})$ and the maximizer $\tilde{\boldsymbol{\eta}}$ in Steps 8 and 9 of Algorithm 3 is $O(T\mathcal{C}_{2g} + T + p^2)$, where T is the number of observed networks. Overall, the complexity of Algorithm 3 is $O\big(C[T(sK\mathcal{C}_\alpha + s\mathcal{C}_{2g} + sp + sp^2) + T\mathcal{C}_{2g} + T + p^2]\big)$, where C is the number of

Algorithm 5 Contrastive Divergence sampling

Input: MCMC transition step K, parameter $\boldsymbol{\eta} = (\boldsymbol{\eta}^+, \boldsymbol{\eta}^-)$, $\{\boldsymbol{y}^{t-1}, \boldsymbol{y}^t\}$
Set $\tilde{\boldsymbol{y}} = \boldsymbol{y}^t$
for $k = 1, \cdots, K$ **do**
    Choose randomly a dyad (i, j) s.t. $i \neq j$
    Propose a dyad value $\tilde{\boldsymbol{y}}_{ij}^{k+1}$ from Equation (4.7)
    Calculate the acceptance ratio α for $\tilde{\boldsymbol{y}}_{ij}^{k+1}$ from Equation (4.8)
    if uniform$(0, 1) < \alpha$ **then**
        Accept $\tilde{\boldsymbol{y}}_{ij}^{k+1}$
    end if
end for
Output: a sampled network $\tilde{\boldsymbol{y}}$

learning iterations. The complexity of Algorithm 4 is similar, except for different choices of the input parameters and MCMC variation in the proposed dyad values.

4.7.4 Experiment Details

In this section, we provide the formulations of selected network statistics used in the two real data examples. The network statistics of interest are chosen from an extensive list in `ergm` [HHB22], an `R` library for network analysis. The choice of network statistics for the increment process is identical to that for the decrement process. Furthermore, we provide the learning schedules of the two experiments.

4.7.4.1 Students Contact Networks

The formulations of six network statistics used for the students contact networks [MFB15] are displayed in Table 4.8. The superscripts are omitted for notational simplicity. In this time-heterogeneous PST ERGM $\prod_{t=2}^{5} P(\boldsymbol{y}^t | \boldsymbol{y}^{t-1}; \boldsymbol{\eta}^t)$, the parameter $\boldsymbol{\eta}^t$ is learned sequentially for

$t = 2, \cdots, 5$, and we initialize $\boldsymbol{\eta}_0^t$ of Algorithm 3 as zero vector.

Network Statistics	Formulations
Edge sum	$\sum_{i<j} \boldsymbol{y}_{ij}$
Dispersion	$\sum_{i<j} \sqrt{\boldsymbol{y}_{ij}}$
Homophily (M)	$\sum_{i<j} \boldsymbol{y}_{ij} \times \mathbb{1}(\boldsymbol{x}_i = \mathrm{M} \wedge \boldsymbol{x}_j = \mathrm{M})$
Heterophily (M-F)	$\sum_{i<j} \boldsymbol{y}_{ij} \times \mathbb{1}(\boldsymbol{x}_i \neq \boldsymbol{x}_j)$
Facebook	$\sum_{i<j} \boldsymbol{y}_{ij} \times \boldsymbol{z}_{ij}$
Transitive weight	$\sum_{i<j} \min(\boldsymbol{y}_{ij}, \max_{k \in N}(\min(\boldsymbol{y}_{ik}, \boldsymbol{y}_{kj})))$

Table 4.8: The network statistics used for the students contact networks.

To seed an initial configuration for each $\boldsymbol{\eta}^t$, we implement $C = 20$ iterations of Algorithm 3 where an MCMC sample size of $s = 100$ with $\mathrm{CD}_{5\times n}$ sampling is used, followed by another $C = 20$ iterations of Algorithm 4 where an MCMC sample size of $s = 100$ with $\mathrm{CD}_{10\times n}$ sampling is used. The term n is the number of students in this data, which is 29. Subsequently, to refine the learned parameters, we implement $C = 10$ iterations of Algorithm 4 where an MCMC sample size of $s = 1000$ with $\mathrm{CD}_{25\times n\times n}$ sampling is used.

Finally, the standard errors are obtained from the Fisher Information matrix $\boldsymbol{I}(\boldsymbol{\eta}^t) \approx -\hat{\boldsymbol{H}}(\tilde{\boldsymbol{\eta}}^t)$ of Equation (4.5) evaluated at the learned parameter $\tilde{\boldsymbol{\eta}}^t$ with 1000 sampled networks. Each sampled network is generated after $K = 20 \times n \times n$ MCMC transitions starting from observed $\boldsymbol{y}^t$.

4.7.4.2 Baboons Interaction Networks

The formulations of four network statistics used for the baboons interaction networks [GGP20] are displayed in Table 4.9. The superscripts are omitted for notational simplicity. In this time-homogeneous PST ERGM $\prod_{t=2}^{23} P(\boldsymbol{y}^t|\boldsymbol{y}^{t-1})$, the parameter $\boldsymbol{\eta}$ is shared across $t = 2, \cdots, 23$ and is also used to forecast the temporal trends for $t = 24, \cdots, 28$. We

initialize $\boldsymbol{\eta}_0$ of Algorithm 3 as zero vector.

Network Statistics	Formulations
Edge sum	$\sum_{i<j} \boldsymbol{y}_{ij}$
Propensity	$\sum_{i<j} \mathbb{1}(\boldsymbol{y}_{ij} > 0)$
Dispersion	$\sum_{i<j} \sqrt{\boldsymbol{y}_{ij}}$
Transitive weight	$\sum_{i<j} \min(\boldsymbol{y}_{ij}, \max_{k \in N}(\min(\boldsymbol{y}_{ik}, \boldsymbol{y}_{kj})))$

Table 4.9: The network statistics used for the baboons interaction networks.

To seed an initial configuration for $\boldsymbol{\eta}$, we implement $C = 20$ iterations of Algorithm 3 where an MCMC sample size of $s = 100$ with CD_n sampling is used for each time t, followed by another $C = 20$ iterations of Algorithm 4 where an MCMC sample size of $s = 100$ with $\text{CD}_{2\times n}$ sampling is used for each time t. The term n is the number of baboons in this data, which is 13. Subsequently, to refine the learned parameter, we implement $C = 10$ iterations of Algorithm 4 where an MCMC sample size of $s = 1000$ with $\text{CD}_{50\times n\times n}$ sampling is used for each time t. In this experiment, we let the maximum dyad value of decrement networks $m^t = 200$ for each t, a moderate upper bound that is greater than the highest dyad value of decrement networks $\boldsymbol{y}^{-,2}, \cdots, \boldsymbol{y}^{-,28}$ constructed from the observed networks.

Finally, the standard errors are obtained from the Fisher Information matrix $\boldsymbol{I}(\boldsymbol{\eta}) \approx -\hat{\boldsymbol{H}}(\tilde{\boldsymbol{\eta}})$ of Equation (4.5) evaluated at the learned parameter $\tilde{\boldsymbol{\eta}}$ with 1000 sampled networks for each time t. Each sampled network is generated after $K = 20 \times n \times n$ MCMC transitions starting from observed $\boldsymbol{y}^t$.

CHAPTER 5

Conclusion

This dissertation presents three self-contained papers, focusing on the statistical models and computational techniques for change point detection and dynamic networks modeling. The extensive experiments on both simulated and real data demonstrate the effectiveness of the proposed methods.

The framework with the Separable Temporal Exponential-family Random Graph Model (STERGM) detects structural changes in network dynamics via network statistics. By employing the Alternating Direction Method of Multipliers (ADMM) and Group Fused Lasso regularization, the learned parameters can reflect multiple time points where the network structures have substantially changed. Yet, this approach requires users to specify the types of structural change to be detected, which are often not known to the modeler before the implementation. Therefore, another approach to extract the change patterns from time series of graphs is in high demand.

The introduction of a generative model presents a different avenue in the change point detection methodologies. By incorporating prior distributions and a graph decoding mechanism, the empirical Bayes approach to learn the priors provides a graph representation learning framework for change point detection. However, several extensions are possible for future developments. Generative models that can produce weighted graphs, as well as nodal and dyadic attributes, should be included. Subsequently, a more complicated neural network architecture is required to permit these extension.

Lastly, the Partially Separable Temporal Exponential-family Random Graph Model (PST

ERGM) focuses on networks whose relations possess degree of strength. The dynamics between consecutive valued networks are decomposed into two intermediate networks, where one controls dyad value increment and the other controls dyad value decrement. The proposed model specifies each transition with two sets of network statistics derived from the intermediate networks and uses two distinct parameters to facilitate model interpretation. Improvements to PST ERGM are also possible for future development. The sample space can be extended to networks with continuous dyad values, since more real-world situations can be included. As time evolve, the number of nodes may not be fixed. It is of great importance for network models to adjust the network sizes accordingly.

The methodologies developed in this dissertation are validated through numeric simulations and real-world applications. The results underscore the potential of these models to provide insights into the dynamics of networks, which can be crucial for decision-making in various domains. In conclusion, this dissertation not only contributes to the field of dynamic networks through statistical modeling and computational techniques, but also sets the stage for future research that could further unravel the complexities of network evolution.

REFERENCES

[ABD13] Carlos M Alaíz, Alvaro Barbero, and José R Dorronsoro. "Group fused lasso." In *Artificial Neural Networks and Machine Learning–ICANN 2013: 23rd International Conference on Artificial Neural Networks Sofia, Bulgaria, September 10-13, 2013. Proceedings 23*, pp. 66–73. Springer, 2013.

[BA18] Leland Bybee and Yves Atchadé. "Change-Point Computation for Large Graphical Models: A Scalable Algorithm for Gaussian Graphical Models with Change-Points." *Journal of Machine Learning Research*, **19**(11):1–38, 2018.

[BB18] Tom Broekel and Marcel Bednarz. "Disentangling link formation and dissolution in spatial networks: An Application of a Two-Mode STERGM to a Project-Based R&D Network in the German Biotechnology Industry." *Networks and Spatial Economics*, **18**(3):677–704, 2018.

[Bes74] Julian Besag. "Spatial interaction and the statistical analysis of lattice systems." *Journal of the Royal Statistical Society: Series B (Methodological)*, **36**(2):192–225, 1974.

[BH22] Bart Blackburn and Mark S Handcock. "Practical Network Modeling via Tapered Exponential-Family Random Graph Models." *Journal of Computational and Graphical Statistics*, pp. 1–14, 2022.

[BLS23] Carter T Butts, Alessandro Lomi, Tom AB Snijders, and Christoph Stadtfeld. "Relational event models in network science." *Network Science*, **11**(2):175–183, 2023.

[BPC11] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. "Distributed optimization and statistical learning via the alternating direction method of multipliers." *Foundations and Trends® in Machine Learning*, **3**(1):1–122, 2011.

[But08a] Carter T Butts. "A relational event framework for social action." *Sociological Methodology*, **38**(1):155–200, 2008.

[But08b] Carter T Butts. "A relational event framework for social action." *Sociological Methodology*, **38**(1):155–200, 2008.

[BV11] Kevin Bleakley and Jean-Philippe Vert. "The group fused lasso for multiple change-point detection." *arXiv preprint arXiv:1106.4199*, 2011.

[BW20] Gerrit JJ van den Burg and Christopher KI Williams. "An evaluation of change point detection algorithms." *arXiv preprint arXiv:2003.06222*, 2020.

[CAA20] Guodong Chen, Jesús Arroyo, Avanti Athreya, Joshua Cape, Joshua T Vogelstein, Youngser Park, Chris White, Jonathan Larson, Weiwei Yang, and Carey E Priebe. "Multiple network embedding for anomaly detection in time series of graphs." *arXiv preprint arXiv:2008.10055*, 2020.

[CC19] Lynna Chu and Hao Chen. "Asymptotic distribution-free change-point detection for multivariate and non-Euclidean data." *The Annals of Statistics*, **47**(1):382–414, 2019.

[CF11] Alberto Caimo and Nial Friel. "Bayesian inference for exponential random graph models." *Social networks*, **33**(1):41–55, 2011.

[CG20] Alberto Caimo and Isabella Gollini. "A multilayer exponential random graph modelling approach for weighted networks." *Computational Statistics & Data Analysis*, **142**:106825, 2020.

[CH05] Miguel A Carreira-Perpinan and Geoffrey Hinton. "On contrastive divergence learning." In *International workshop on Artificial Intelligence and Statistics*, pp. 33–40. PMLR, 2005.

[Che19] Hao Chen. "Sequential change-point detection based on nearest neighbors." *The Annals of Statistics*, **47**(3):1381–1407, 2019.

[CZ15] Hao Chen and Nancy Zhang. "Graph-based change-point detection." *The Annals of Statistics*, **43**(1):139–176, 2015.

[CZC20] Hao Chen, Nancy R. Zhang, Lynna Chu, and Hoseung Song. *gSeg: Graph-Based Change-Point Detection (g-Segmentation)*, 2020. R package version 1.0.

[DC12a] Bruce A Desmarais and Skyler J Cranmer. "Statistical inference for valued-edge networks: The generalized exponential random graph model." *PloS one*, **7**(1):e30136, 2012.

[DC12b] Bruce A Desmarais and Skyler J Cranmer. "Statistical mechanics of networks: Estimation and uncertainty." *Physica A: Statistical Mechanics and its Applications*, **391**(4):1865–1876, 2012.

[DCL18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805*, 2018.

[DH18] Claire Donnat and Susan Holmes. "Tracking network dynamics: A survey using graph distances." *The Annals of Applied Statistics*, **12**(2):971–1012, 2018.

[DLL21] Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. "Graph neural networks with learnable structural and positional representations." *arXiv preprint arXiv:2110.07875*, 2021.

[DYP21] Jiaying Deng, Mingwen Yang, M. Pelster, and Yong Tan. "A Boon or a Bane? An Examination of Social Communication in Social Trading." *Capital Markets: Market Efficiency eJournal*, 2021.

[EP06] Nathan Eagle and Alex (Sandy) Pentland. "Reality mining: sensing complex social systems." *Personal and Ubiquitous Computing*, **10**(4):255–268, 2006.

[FDR21] Cornelius Fritz, Emilio Dorigatti, and David Rügamer. "Combining graph neural networks and spatio-temporal disease models to predict covid-19 cases in germany." *arXiv preprint arXiv:2101.00661*, 2021.

[Fel14] Ian E Fellows. "Why (and when and how) contrastive divergence works." *arXiv preprint arXiv:1405.0602*, 2014.

[FH12a] Ian Fellows and Mark S. Handcock. "Exponential-family Random Network Models." *arXiv preprint arxiv:1208.0121*, 2012.

[FH12b] Ian Fellows and Mark S. Handcock. "Exponential-family Random Network Models.", 2012.

[FH13] Ian E Fellows and Mark S Handcock. "Analysis of partially observed networks via exponential-family random network models." *arXiv preprint arXiv:1303.1219*, 2013.

[FH17] Ian Fellows and Mark Handcock. "Removing phase transitions from Gibbs measures." In *Artificial intelligence and statistics*, pp. 289–297. PMLR, 2017.

[FHW19] Fuli Feng, Xiangnan He, Xiang Wang, Cheng Luo, Yiqun Liu, and Tat-Seng Chua. "Temporal relational ranking for stock prediction." *ACM Transactions on Information Systems (TOIS)*, **37**(2):1–30, 2019.

[GA18] Damien Garreau and Sylvain Arlot. "Consistent change-point detection with kernels." *Electronic Journal of Statistics*, **12**(2):4440 – 4486, 2018.

[GD20] Ravi Goyal and Victor De Gruttola. "Dynamic network prediction." *Network Science*, **8**(4):574–595, 2020.

[GDZ23] Yongshun Gong, Xue Dong, Jian Zhang, and Meng Chen. "Latent evolution model for change point detection in time-varying networks." *Information Sciences*, **646**:119376, 2023.

[GGP20] Valeria Gelardi, Jeanne Godard, Dany Paleressompoulle, Nicolas Claidière, and Alain Barrat. "Measuring social networks in primates: wearable sensors versus direct observations." *Proceedings of the Royal Society A*, **476**(2236):20190737, 2020.

[GT92] Charles J Geyer and Elizabeth A Thompson. "Constrained Monte Carlo maximum likelihood for dependent data." *Journal of the Royal Statistical Society: Series B (Methodological)*, **54**(3):657–683, 1992.

[HBS23] Mingqi Han, Eric A Bushong, Mayuko Segawa, Alexandre Tiard, Alex Wong, Morgan R Brady, Milica Momcilovic, Dane M Wolf, Ralph Zhang, Anton Petcherski, et al. "Spatial mapping of mitochondrial networks and bioenergetics in lung cancer." *Nature*, **615**(7953):712–719, 2023.

[HFX10] Steve Hanneke, Wenjie Fu, and Eric P Xing. "Discrete temporal models of social networks." *Electronic Journal of Statistics*, **4**:585–605, 2010.

[HGH08] David R Hunter, Steven M Goodreau, and Mark S Handcock. "Goodness of fit of social network models." *Journal of the American Statistical Association*, **103**(481):248–258, 2008.

[HH06a] David R Hunter and Mark S Handcock. "Inference in curved exponential family models for networks." *Journal of Computational and Graphical Statistics*, **15**(3):565–583, 2006.

[HH06b] David R Hunter and Mark S Handcock. "Inference in curved exponential family models for networks." *Journal of Computational and Graphical Statistics*, **15**(3):565–583, 2006.

[HHB08a] David R. Hunter, Mark S. Handcock, Carter T. Butts, Steven M. Goodreau, and Martina Morris. "ergm: A Package to Fit, Simulate and Diagnose Exponential-Family Models for Networks." *Journal of Statistical Software*, **24**(3):1–29, 2008.

[HHB08b] David R Hunter, Mark S Handcock, Carter T Butts, Steven M Goodreau, and Martina Morris. "ergm: A package to fit, simulate and diagnose exponential-family models for networks." *Journal of Statistical Software*, **24**(3):nihpa54860, 2008.

[HHB22] Mark S. Handcock, David R. Hunter, Carter T. Butts, Steven M. Goodreau, Pavel N. Krivitsky, and Martina Morris. *ergm: Fit, Simulate and Diagnose Exponential-Family Models for Networks*. The Statnet Project (`https://statnet.org`), 2022. R package version 4.3.2.

[HHH12a] Ruth M Hummel, David R Hunter, and Mark S Handcock. "Improving simulation-based algorithms for fitting ERGMs." *Journal of Computational and Graphical Statistics*, **21**(4):920–939, 2012.

[HHH12b] Ruth M Hummel, David R Hunter, and Mark S Handcock. "Improving simulation-based algorithms for fitting ERGMs." *Journal of Computational and Graphical Statistics*, **21**(4):920–939, 2012.

[HHL23] Xiaoxin He, Bryan Hooi, Thomas Laurent, Adam Perold, Yann LeCun, and Xavier Bresson. "A generalization of vit/mlp-mixer to graphs." In *International Conference on Machine Learning*, pp. 12724–12745. PMLR, 2023.

[HHR20] Shenyang Huang, Yasmeen Hitti, Guillaume Rabusseau, and Reihaneh Rabbany. "Laplacian change point detection for dynamic graphs." In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 349–358, 2020.

[Hin02] Geoffrey E Hinton. "Training products of experts by minimizing contrastive divergence." *Neural Computation*, **14**(8):1771–1800, 2002.

[HJA20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." *Advances in neural information processing systems*, **33**:6840–6851, 2020.

[HRS03] Mark S Handcock, Garry Robins, Tom Snijders, Jim Moody, and Julian Besag. "Assessing degeneracy in statistical models of social networks." Technical report, Working paper, 2003.

[HRT07] Mark S Handcock, Adrian E Raftery, and Jeremy M Tantrum. "Model-based clustering for social networks." *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, **170**(2):301–354, 2007.

[Hum11] Ruth M Hummel. *Improving estimation for exponential-family random graph models*. PhD thesis, The Pennsylvania State University, 2011.

[JLY20] Binyan Jiang, Jailing Li, and Qiwei Yao. "Autoregressive networks." *arXiv preprint arXiv:2010.04492*, 2020.

[JM15] Nicholas A. James and David S. Matteson. "ecp: An R Package for Nonparametric Multiple Change Point Analysis of Multivariate Data." *Journal of Statistical Software*, **62**(7):1–25, 2015.

[KCP23] Yik Lun Kei, Yanzhen Chen, and Oscar Hernan Madrid Padilla. "A Partially Separable Model for Dynamic Valued Networks." *Computational Statistics & Data Analysis*, p. 107811, 2023.

[KH14] Pavel N Krivitsky and Mark S Handcock. "A separable model for dynamic networks." *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, **76**(1):29, 2014.

[KH22] Pavel N. Krivitsky and Mark S. Handcock. *tergm: Fit, Simulate and Diagnose Models for Network Evolution Based on Exponential-Family Random Graph Models*. The Statnet Project (`https://statnet.org`), 2022. R package version 4.1.0.

[KHM11] Pavel N Krivitsky, Mark S Handcock, and Martina Morris. "Adjusting for network size and composition effects in exponential-family random graph models." *Statistical Methodology*, **8**(4):319–339, 2011.

[KLC23] Yik Lun Kei, Hangjian Li, Yanzhen Chen, and Oscar Hernan Madrid Padilla. "Change Point Detection on a Separable Model for Dynamic Networks." *arXiv preprint arXiv:2303.17642*, 2023.

[KLL24] Yik Lun Kei, Jialiang Li, Hangjian Li, Yanzhen Chen, and Oscar Hernan Madrid Padilla. "Change Point Detection in Dynamic Graphs with Generative Model." *arXiv preprint arXiv:2404.04719*, 2024.

[Kri12a] Pavel N Krivitsky. "Exponential-family random graph models for valued networks." *Electronic Journal of Statistics*, **6**:1100, 2012.

[Kri12b] Pavel N Krivitsky. "Exponential-family random graph models for valued networks." *Electronic journal of statistics*, **6**:1100, 2012.

[Kri17a] Pavel N Krivitsky. "Using contrastive divergence to seed Monte Carlo MLE for exponential-family random graph models." *Computational Statistics & Data Analysis*, **107**:149–161, 2017.

[Kri17b] Pavel N Krivitsky. "Using contrastive divergence to seed Monte Carlo MLE for exponential-family random graph models." *Computational Statistics & Data Analysis*, **107**:149–161, 2017.

[Kri19] Pavel N. Krivitsky. *ergm.count: Fit, Simulate and Diagnose Exponential-Family Models for Networks with Count Edges*. The Statnet Project (`https://statnet.org`), 2019. R package version 3.4.0.

[KSA10] Mladen Kolar, Le Song, Amr Ahmed, and Eric P Xing. "Estimating time-varying networks." *The Annals of Applied Statistics*, pp. 94–123, 2010.

[LBF21] Federico Larroca, Paola Bermolen, Marcelo Fiori, and Gonzalo Mateos. "Change Point Detection in Weighted and Directed Random Dot Product Graphs." In *2021 29th European Signal Processing Conference (EUSIPCO)*, pp. 1810–1814. IEEE, 2021.

[LCX18] Fuchen Liu, David Choi, Lu Xie, and Kathryn Roeder. "Global spectral clustering in dynamic networks." *Proceedings of the National Academy of Sciences*, **115**(5):927–932, 2018.

[LEN18] Matthew Ludkin, Idris Eckley, and Peter Neal. "Dynamic stochastic block models: parameter estimation and detection of changes in community structure." *Statistics and Computing*, **28**(6):1201–1213, 2018.

[LH07] Céline Levy-leduc and Zaïd Harchaoui. "Catching Change-points with Lasso." In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007.

[LLG19] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension." *arXiv preprint arXiv:1910.13461*, 2019.

[LRS18] Steffen Lauritzen, Alessandro Rinaldo, and Kayvan Sadeghi. "Random networks, graphical models and exchangeability." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **80**(3):481–508, 2018.

[MBF22] Bernardo Marenco, Paola Bermolen, Marcelo Fiori, Federico Larroca, and Gonzalo Mateos. "Online Change Point Detection for Weighted and Directed Random Dot Product Graphs." *IEEE Transactions on Signal and Information Processing over Networks*, **8**:144–159, 2022.

[MFB15] Rossana Mastrandrea, Julie Fournet, and Alain Barrat. "Contact patterns in a high school: a comparison between data collected using wearable sensors, contact diaries and friendship surveys." *PloS one*, **10**(9):e0136497, 2015.

[MHH08] Martina Morris, Mark S Handcock, and David R Hunter. "Specification of exponential-family random graph models: terms and computational aspects." *Journal of statistical software*, **24**(4):1548, 2008.

[MM17] Catherine Matias and Vincent Miele. "Statistical clustering of temporal networks through a dynamic stochastic block model." *Journal of the Royal Statistical Society Series B: Statistical Methodology*, **79**(4):1119–1141, 2017.

[MXW23] Carlos Misael Madrid Padilla, Haotian Xu, Daren Wang, Oscar Hernan Madrid Padilla, and Yi Yu. "Change point detection and inference in multivariable nonparametric models under mixing conditions." *arXiv preprint arXiv:2301.11491*, 2023.

[MYP22] Oscar Hernan Madrid Padilla, Yi Yu, and Carey E Priebe. "Change point localization in dependent dynamic nonparametric random dot product graphs." *The Journal of Machine Learning Research*, **23**(1):10661–10719, 2022.

[MYW21] Oscar Hernan Madrid Padilla, Yi Yu, Daren Wang, and Alessandro Rinaldo. "Optimal nonparametric multivariate change point detection and localization." *IEEE Transactions on Information Theory*, **68**(3):1922–1944, 2021.

[NHH20] Erik Nijkamp, Mitch Hill, Tian Han, Song-Chun Zhu, and Ying Nian Wu. "On the anatomy of mcmc-based maximum likelihood learning of energy-based models."

In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5272–5280, 2020.

[OOC21] Martin Ondrus, Emily Olds, and Ivor Cribben. "Factorized Binary Search: change point detection in the network structure of multivariate high-dimensional time series." *arXiv preprint arXiv:2103.06347*, 2021.

[PC15] Leto Peel and Aaron Clauset. "Detecting change points in the large-scale structure of evolving networks." In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[PCM05] Carey E Priebe, John M Conroy, David J Marchette, and Youngser Park. "Scan statistics on enron graphs." *Computational & Mathematical Organization Theory*, **11**:229–247, 2005.

[Pen19] Marianna Pensky. "Dynamic network models and graphon estimation." *The Annals of Statistics*, **47**(4):2378–2403, 2019.

[PHN20] Bo Pang, Tian Han, Erik Nijkamp, Song-Chun Zhu, and Ying Nian Wu. "Learning latent space energy-based prior model." *Advances in Neural Information Processing Systems*, **33**:21994–22008, 2020.

[PPY12] Youngser Park, Carey E Priebe, and Abdou Youssef. "Anomaly detection in time series of graphs using fusion of graph invariants." *IEEE journal of selected topics in signal processing*, **7**(1):67–75, 2012.

[PS20] Jong Hee Park and Yunkyu Sohn. "Detecting Structural Changes in Longitudinal Network Data." *Bayesian Analysis*, **15**(1):133 – 157, 2020.

[PYP19] Oscar Hernan Madrid Padilla, Yi Yu, and Carey E Priebe. "Change point localization in dependent dynamic nonparametric random dot product graphs." *arXiv preprint arXiv:1911.07494*, 2019.

[PYP22] Oscar Hernan Madrid Padilla, Yi Yu, and Carey E. Priebe. "Change point localization in dependent dynamic nonparametric random dot product graphs." *Journal of Machine Learning Research*, **23**(234):1–59, 2022.

[RBL22] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. "High-resolution image synthesis with latent diffusion models." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

[RP01] Garry Robins and Philippa Pattison. "Random graph models for temporal processes in social networks." *Journal of Mathematical Sociology*, **25**(1):5–41, 2001.

[RPW09] Garry Robins, Pip Pattison, and Peng Wang. "Closure, connectivity and degree distributions: Exponential random graph (p*) models for directed social networks." *Social Networks*, **31**(2):105–117, 2009.

[SBS10] Tom AB Snijders, Gerhard G Van de Bunt, and Christian EG Steglich. "Introduction to stochastic actor-based models for network dynamics." *Social networks*, **32**(1):44–60, 2010.

[SC15] Daniel K Sewell and Yuguo Chen. "Latent space models for dynamic networks." *Journal of the American Statistical Association*, **110**(512):1646–1657, 2015.

[SC16] Daniel K Sewell and Yuguo Chen. "Latent space models for dynamic networks with weighted edges." *Social Networks*, **44**:105–116, 2016.

[SC22a] Hoseung Song and Hao Chen. "Asymptotic distribution-free changepoint detection for data with repeated observations." *Biometrika*, **109**(3):783–798, 2022.

[SC22b] Hoseung Song and Hao Chen. *kerSeg: New Kernel-Based Change-Point Detection*, 2022. R package version 1.0.

[SC22c] Hoseung Song and Hao Chen. "New kernel-based change-point detection." *arXiv preprint arXiv:2206.01853*, 2022.

[SI90] David Strauss and Michael Ikeda. "Pseudolikelihood estimation for social networks." *Journal of the American Statistical Association*, **85**(409):204–212, 1990.

[SKC23] Deborah Sulem, Henry Kenlay, Mihai Cucuringu, and Xiaowen Dong. "Graph similarity learning for change-point detection in dynamic networks." *Machine Learning*, pp. 1–44, 2023.

[SM05] Purnamrita Sarkar and Andrew W Moore. "Dynamic social network analysis using latent space models." *ACM SIGKDD Explorations Newsletter*, **7**(2):31–40, 2005.

[SM17] David R Schaefer and Christopher Steven Marcum. "Modeling network dynamics." In *The Oxford handbook of social networks*, pp. 254–287. Oxford University Press New York, 2017.

[SMS21] Mohammed Suhail, Abhay Mittal, Behjat Siddiquie, Chris Broaddus, Jayan Eledath, Gerard Medioni, and Leonid Sigal. "Energy-based learning for scene graph generation." In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13936–13945, 2021.

[Sni01] Tom AB Snijders. "The statistical evaluation of social network dynamics." *Sociological Methodology*, **31**(1):361–395, 2001.

[Sni02] Tom AB Snijders. "Markov chain Monte Carlo estimation of exponential random graph models." *Journal of Social Structure*, **3**(2):1–40, 2002.

[Sni05] Tom AB Snijders. "Models for longitudinal network data." *Models and Methods in Social Network Analysis*, **1**:215–247, 2005.

[SPR06] Tom AB Snijders, Philippa E Pattison, Garry L Robins, and Mark S Handcock. "New specifications for exponential random graph models." *Sociological methodology*, **36**(1):99–153, 2006.

[SS22] S Golshid Sharifnia and Abbas Saghaei. "A statistical approach for social network change detection: an ERGM based framework." *Communications in Statistics-Theory and Methods*, **51**(7):2259–2280, 2022.

[TB11] Andrew C Thomas and Joseph K Blitzstein. "Valued ties tell fewer lies: Why not to dichotomize network edges with thresholds." *arXiv preprint arXiv:1101.0788*, 2011.

[TFC16] Stephanie Thiemichen, Nial Friel, Alberto Caimo, and Göran Kauermann. "Bayesian exponential random graph models with nodal random effects." *Social Networks*, **46**:11–28, 2016.

[UH20] Medha Uppala and Mark S Handcock. "Modeling wildfire ignition origins in southern California using linear network point processes." *The Annals of Applied Statistics*, **14**(1):339–356, 2020.

[VB10] Jean-Philippe Vert and Kevin Bleakley. "Fast detection of multiple change-points shared by many signals using group LARS." *Advances in Neural Information Processing Systems*, **23**, 2010.

[VGH09] Marijtje AJ Van Duijn, Krista J Gile, and Mark S Handcock. "A framework for the comparison of maximum pseudo-likelihood and maximum likelihood estimation of exponential family random graph models." *Social Networks*, **31**(1):52–62, 2009.

[WCB10] Danny Wyatt, Tanzeem Choudhury, and Jeff Bilmes. "Discovering long range properties of social networks with multi-valued time-inhomogeneous models." In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pp. 630–636, 2010.

[WDB17] James D Wilson, Matthew J Denny, Shankar Bhamidi, Skyler J Cranmer, and Bruce A Desmarais. "Stochastic weighted graphs: Flexible model specification and simulation." *Social Networks*, **49**:37–47, 2017.

[WP96] Stanley Wasserman and Philippa Pattison. "Logit models and logistic regressions for social networks: I. An introduction to Markov graphs andp." *Psychometrika*, **61**(3):401–425, 1996.

[WTP13] Heng Wang, Minh Tang, Youngser Park, and Carey E Priebe. "Locality statistics for anomaly detection in time series of graphs." *IEEE Transactions on Signal Processing*, **62**(3):703–717, 2013.

[WYR21] Daren Wang, Yi Yu, and Alessandro Rinaldo. "Optimal change point detection and localization in sparse dynamic networks." *The Annals of Statistics*, **49**(1):203–232, 2021.

[XBS20] Jian Xie, Youyi Bi, Zhenghui Sha, Mingxian Wang, Yan Fu, Noshir Contractor, Lin Gong, and Wei Chen. "Data-Driven Dynamic Network Modeling for Analyzing the Evolution of Product Competitions." *Journal of Mechanical Design*, **142**(3), 2020.

[XLG18] Jianwen Xie, Yang Lu, Ruiqi Gao, Song-Chun Zhu, and Ying Nian Wu. "Cooperative training of descriptor and generator networks." *IEEE transactions on pattern analysis and machine intelligence*, **42**(1):27–45, 2018.

[XZN17] Jianwen Xie, Song-Chun Zhu, and Ying Nian Wu. "Synthesizing dynamic patterns by spatial-temporal generative convnet." In *Proceedings of the ieee conference on computer vision and pattern recognition*, pp. 7093–7101, 2017.

[YCZ11] Tianbao Yang, Yun Chi, Shenghuo Zhu, Yihong Gong, and Rong Jin. "Detecting communities and their evolutions in dynamic social networks—a Bayesian approach." *Machine learning*, **82**:157–189, 2011.

[YL06] Ming Yuan and Yi Lin. "Model selection and estimation in regression with grouped variables." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **68**(1):49–67, 2006.

[YMW21] Yi Yu, Oscar Hernan Madrid Padilla, Daren Wang, and Alessandro Rinaldo. "Optimal network online change point localisation." *arXiv preprint arXiv:2101.05477*, 2021.

[YSH21] George G Vega Yon, Andrew Slaughter, and Kayla de la Haye. "Exponential random graph models for little networks." *Social Networks*, **64**:225–238, 2021.

[ZCL19] Zifeng Zhao, Li Chen, and Lizhen Lin. "Change-point detection in dynamic networks via graphon estimation." *arXiv preprint arXiv:1908.01823*, 2019.

[ZDP19] Chen Zhang, Xinghua Dang, Tao Peng, and Chaokai Xue. "Dynamic evolution of venture capital network in clean energy industries based on STERGM." *Sustainability*, **11**(22):6313, 2019.